AD-A031 209 UNITED TECHNOLOGIES RESEARCH CENTER EAST HARTFORD CONN F/G 20/4
A COORDINATE SYSTEM FOR A VISCOUS TRANSONIC CASCADE ANALYSIS.(U)
MAY 76 P R EISEMAN N00019-75-C-0463
UNCLASSIFIED UTRC/R76-912149-4 NL

1 OF 1
AD A031209

END

DATE FILMED
11-76

R76-912149-4

A COORDINATE SYSTEM FOR A VISCOUS TRANSONIC CASCADE ANALYSIS

(12)

Peter R. Eiseman
United Technologies Research Center
East Hartford, Connecticut 06108

11 May 1976

Final report for period
12 May 1975 - 11 May 1976

Approved for public release;
Distribution unlimited

(9)

Prepared for

NAVAL AIR SYSTEMS COMMAND
Department of the Navy
1499 Jefferson Davis Hwy.
Arlington, Va. 22209

UNITED TECHNOLOGIES RESEARCH CENTER
EAST HARTFORD CONNECTICUT 06108

76-05-48-1

(9) Final rept. 12 May 75 - 11 May 76

MIL-STD-847A
31 January 1973

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: R76-912149-4	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A Coordinate System For A Viscous Transonic Cascade Analysis (6)	5. TYPE OF REPORT & PERIOD COVERED: Final Report, May 1975-May 1976
(14) UTRC Report R76-912149-4	
7. AUTHOR(s): (10) Peter R. Eiseman	8. CONTRACT OR GRANT NUMBER(s): N00019-75-C-0463
9. PERFORMING ORGANIZATION NAME AND ADDRESS: United Technologies Research Center, Silver Lane, East Hartford, CT 06108 (15)	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (12) 56 p.
11. CONTROLLING OFFICE NAME AND ADDRESS: (11) (11)	12. REPORT DATE: May [illegible]976
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Department of the Navy, Naval Air Systems Command, Arlington, VA 22217	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Least Squares Spline
Cascade Coordinates
Metric Tensor
Navier-Stokes Equations
Reparameterization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A coordinate system suitable for the numerical computation of viscous transonic cascade flows is constructed. The general cascade problem is discussed, and solution of the Navier-Stokes equations is determined to be an effective method of analysis for this problem. The representation of cascade geometries is discussed, and the use of coordinate transformation is adopted.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409252

Many of the difficulties associated with previous geometric representations are overcome with the construction of a new coordinate system which is especially tailored to the numerical simulation of viscous flows through a cascade of airfoils. The system consists of coordinate loops surrounding the airfoil and radial coordinate lines normal to the airfoil surface. The outermost loop is constructed so that the cascade periodicity conditions can be applied without interpolation between grid points. The coordinates are orthogonal on the airfoil surface but gradually become nonorthogonal away from the airfoil. Large gradients in the viscous shear layers are adequately resolved with a simple coordinate distribution function along the outward normal direction from the airfoil. An essential ingredient in the generation of the cascade coordinates is the conversion of discrete curve-like data into analytically defined curves which accurately reflect the overall geometry. For this purpose a least squares spline procedure is employed. The coordinate generation procedure accepts discrete input data representing the airfoil surface, places little restriction on the airfoil camber or spacing and is easily extended to three dimensions. The coordinate system is thus more general and in many cases computationally less complex than systems derived from conformal transformations.

A Coordinate System For A Viscous Transonic Cascade Analysis

TABLE OF CONTENTS

I. INTRODUCTION

An important problem which must be faced by the designer of advanced gas turbine engines is the prediction of the flow field in and around turbine and compressor blade passages. An accurate estimate of the flow field is required to predict the heat transfer rates and aerodynamic losses both of which may be critical to successful engine operation. In advanced engines, it is expected that turbine and compressor blade passages may contain transonic flow and in these more complex transonic flow cases, techniques for predicting heat transfer rates and aerodynamic losses in the transonic regime would provide a valuable tool in the engine design process. In this regard poor estimates of either loss coefficients or heat transfer may result in poor predictions of engine performance or catastrophic failure of the engine components. For example, excessive heat transfer rates associated with boundary layer separation and reattachment on turbine blades and end walls can have damaging effects as the resulting hot spots may result in structural failure. In addition, excess aerodynamic losses associated with viscous effects may result in a serious deterioration of component efficiency. Since aerodynamic losses and heat transfer rates are associated with the viscous nature of the fluid, the ability to predict the viscous flow in high performance turbine and compressor blade passages becomes quite important to the successful design process. Most presently available analyses of the blade passage flow field have been based upon solutions of the inviscid equations and then corrected for viscous effects through empirical data correlations.

Obviously, methods which completely neglect viscous effects or methods which include viscous effects through empirical corrections are inherently limited. Other more rigorous studies obtain an inviscid flow field and then input the blade pressure distribution into a boundary layer procedure to calculate the viscous boundary layer in the immediate vicinity of the blade. When the boundary layer remains small throughout the entire blade passage, such a boundary layer correction may give an adequate description of the flow even when viscous displacement effects upon the inviscid flow are neglected. However, severe pressure gradients and shock waves can cause boundary layers to thicken or become separated and in such a situation the boundary layer displacement effects are expected to exert a significant influence on the nominally inviscid flow field. When viscous displacement effects do alter the nominally inviscid flow field significantly, it is necessary for the calculation procedure to recognize the mutual dependence between the viscous and inviscid flows either through a strong-interaction analysis between a viscous boundary layer solution and an inviscid outer flow field solution or through a full Navier-Stokes solution in the entire region of interest. In regard to strong interaction analyses, if the inviscid flow is supersonic and if a true inviscid core is present in the blade passage flow field, the governing inviscid equations are hyperbolic in nature, and the interaction analysis can be solved with a forward-marching procedure in which inviscid and viscous regions are coupled implicitly. However, this interaction formulation to a supersonic problem yields a stiff system of partial differential

equations which may be difficult to analyze numerically. If the inviscid flow is subsonic, the equations governing the inviscid flow field are elliptic and hence cannot be solved by a forward-marching procedure in space. In this case a sequence of inviscid and boundary layer solutions must be performed so that each corrects its predecessor until a desired stage of convergence is obtained through a global iteration. In any case, the strong-interaction analysis presents serious difficulties, particularly, in interacting the viscous and inviscid solutions around the airfoils which form the blade passage or in calculating substantial regions of separated flow. Finally, the interaction approach is not even valid for flows in which the viscous region encompasses most of the flow field since, in this case, no nominally inviscid flow region exists.

Due to the limitations and difficulties associated with strong-interaction analyses, particularly in transonic flow with its extreme sensitivity to cross sectional area, a computational method based upon solving the time-dependent Navier-Stokes equations in the entire flow regime would be a favored alternative solution procedure for the viscous blade passage problem. In a Navier-Stokes solution, boundary layer separation and reattachment would evolve naturally with the advantageous use of the same basic numerical analysis for viscous and inviscid regions. In addition, a solution based upon the Navier-Stokes equations would handle shock wave-boundary layer interactions in a natural manner. Solution of the Navier-Stokes equations, however, first requires a coordinate system appropriate for the geometry and boundary conditions of interest. A coordinate system especially tailored for the geometry and periodicity requirements of airfoil cascades is developed and presented herein.

II. BACKGROUND

A computer code capable of accurately predicting transonic flow through a blade passage obviously would have a significant positive impact upon the gas turbine design process as it would aid greatly in obtaining accurate predictions of blade heat transfer and aerodynamic losses. To date most transonic analyses have concentrated upon inviscid solutions of the isolated airfoil problem; a recent review of inviscid isolated airfoil transonic flow analyses has been presented by Yoshihara (Ref. 1). In contrast several inviscid analyses have been developed specifically for the cascade problem. These cascade analyses include the transonic procedures of Delaney and Kavanagh (Ref. 2) and Gopalakrishnan and Bozzola (Ref. 3) among others. Although these inviscid flow analyses can serve to predict certain features of the passage flow, their inviscid assumptions lead to several major limitations.

The major limitations which evolve from an inviscid flow analysis concern application of the Kutta condition, neglect of viscous phenomena, prediction of shock location and predictions of the trailing edge base pressure. Considering first the Kutta condition problem, if the flow field is assumed inviscid, only the normal velocity at the airfoil surface is set to be zero; no restriction is placed upon the tangential velocity. In this situation, the oncoming flow impinges upon the leading edge region of the airfoil and somewhere in this region a leading edge stagnation point appears. The inviscid flow divides at the stagnation point as some of the flow (that which is above the stagnation streamline) proceeds along the upper surface of the airfoil and the remainder of the flow (that below the stagnation streamline) proceeds along the lower surface. Since the flow is inviscid, there is no restriction on the tangential velocity along either surface. This slip velocity presents no problem until the trailing edge is reached. At the trailing edge, the flow along the upper surface rejoins that along the lower surface and passes into the wake; however, there is no guarantee that when the upper surface and lower surface flows join at the trailing edge they will both have identical tangential velocities. In general, the tangential velocities predicted by an inviscid theory will be different and an unrealistic velocity discontinuity will result from an inviscid analysis in the wake. The usual method by which this unrealistic behavior is suppressed is through the application of a Kutta condition in which an airfoil circulation is specified which either makes the airfoil trailing edge a stagnation point or which causes an identical nonzero tangential velocity to appear in both the upper and lower stream at the trailing edge. While the proper specification of the Kutta condition is both reasonable and straightforward for

inviscid flow is calculated ignoring any displacement effects of the blade boundary layer and then a boundary layer calculation is made under the influence of the inviscid pressure distribution. Under severe pressure gradients or at low or moderate Reynolds numbers the boundary layer can thicken or become separated and in such cases the nominally inviscid flow field calculated in the absence of any viscous effects is considerably different from that which is actually present. In the presence of such viscous displacement effects an accurate calculation procedure must recognize the mutual dependence between the viscous and nominally inviscid flow either through a strong interaction analysis (e.g., Erdos, Baronti, and Elzweig; Ref. 8) or a full Navier-Stokes solution in the entire region of interest.

A strong interaction analysis may take the form of either a forward marching procedure or a global iteration. For regions where the inviscid flow is supersonic and thus described by hyperbolic equations, a solution can be marched with the inviscid and viscous regions coupled on a station by station basis. The chief difficulty in this process is that stiff equations must be solved. Common problems with stiff equations show up in the form of numerical solutions which can quickly branch off the desired solution thus producing a physically unrealistic result. In regions where the inviscid flow is subsonic and thus described by elliptic equations, a forward-marching procedure is impossible and consequently a sequence of inviscid and boundary layer solutions must be performed in a manner where each stage corrects the former one through a global iteration. Although transonic inviscid analyses of cascades could be extended by incorporation into a strong-interaction calculation, implementation of the strong-interaction analysis is difficult; and for the transonic cascades of interest the strong-interaction analysis is not expected to be more economical than a full Navier-Stokes solution. Furthermore, the interaction analysis is invalid for flows which are almost entirely viscous; in such flows no inviscid core exists and under these circumstances where no inviscid core exists a reliable solution must be based upon the Navier-Stokes equations. Finally, if an interaction procedure is to be used, the viscous layer is solved by a forward marching boundary layer calculation procedure. In the case of steady state boundary layer procedures, problems will be encountered when the boundary layer is subjected to a strong enough adverse pressure gradient to cause separation. Although a boundary layer procedure can be marched through separation by the usual method of suppressing streamwise convection terms in the separated region (e.g. Ref. 9) the resulting solution is based upon an approximation made in the separated flow region and calculated details of the flow in this region must be viewed with caution.

Since a strong interaction analysis may not be valid in certain cases of interest due to the lack of any identifiable inviscid core and since an interaction calculation requires a time-consuming iteration, a computational method based on solving the time-dependent Navier-Stokes equations is an attractive alternative. With a very competitive computational speed a Navier-Stokes method would be directly extendable to three dimensions and would correctly produce the viscous behavior.

isolated airfoils at high Reynolds numbers, proper specification may not be obvious in cascades for either time-dependent or steady state flows. Any uncertainty in the Kutta condition can be very detrimental to an inviscid flow calculation since even modest modifications in the Kutta condition can lead to significant modifications in the generated potential flow. In contrast to the necessity of applying a Kutta condition in inviscid flow, no such procedure is required in a viscous flow calculation. The viscous flow calculation applies a no-slip boundary condition to the blade surface and, thus, no discontinuity develops. In a viscous flow solution, no circulation is applied but rather the circulation emerges as part of the viscous solution.

A second deficiency associated with an inviscid solution concerns the neglect of viscous phenomena. If viscous phenomena are neglected, aerodynamic losses cannot be predicted; and hence, engine performance cannot be reliably estimated. In addition to this inability to predict aerodynamic loss, an inviscid solution cannot predict wall heat transfer rates and, thus, an inviscid procedure cannot be used to determine the location of hot spots such as may occur at a separated boundary layer reattachment point and cannot predict the cooling required to maintain structural integrity. Obviously, accurate predictions of wall heating and aerodynamic losses are critical to a successful design. In addition, an inviscid calculation may not be adequate for predicting the location of a transonic shock. Since in transonic flow an inviscid calculation is extremely sensitive to the effective airfoil shape, an inviscid determination of the shock location may be unreliable unless the boundary layer thickness is very small. When the physical boundary layer viscous displacement effects change the effective airfoil geometry, the inviscid placement of shocks may be in considerable error. Finally, in the case of a blade with a significant base region, inviscid solutions in general will be inadequate in predicting the base pressure and thus will not correctly account for a significant loss mechanism.

Despite inherent limitations of inviscid flow theory, many existing inviscid transonic analyses have been developed for both the isolated airfoil problem and the cascade problem. These include the cascade analyses of Refs. 2 and 3 and the isolated airfoil analyses presented in Refs. 4-7. All these procedures have met with varying degrees of success for a variety of problems but all are limited by the aforementioned restrictions inherent in the inviscid assumption. In certain cases these limitations could be relieved by coupling a boundary layer computation to the inviscid flow analysis and in all cases the limitions could be relieved by solving the entire flow field with the Navier-Stokes equations.

In cases where the boundary layer is thin, the viscous displacement effects on the nominally inviscid flow field can often be neglected and then it is possible that a suitable prediction of viscous effects may be obtained from simple boundary layer theory. Such an approach has been used to modify some inviscid procedures so as to include viscous effects (e.g., Korn and Garabedian). In those procedures the

Implicit procedures studied by Briley and McDonald at UTRC have led to the development of a highly efficient time-dependent implicit Navier-Stokes computer code which could form a basis for solving the blade pressure problem (Ref. 10).

Navier-Stokes solutions can be obtained either through a relaxation of the steady-state Navier-Stokes equations or through application of the time-dependent Navier-Stokes equations subjected to steady-state boundary conditions. In this latter case, the time-dependent solution would progress to a steady-state and time-steps can be regarded as an iteration. Solution procedures for the time-dependent equations may be either explicit or implicit. If the solution procedure is explicit, the maximum time-step is governed by stringent stability limits (Ref. 11) which relate the maximum time-step to the size of the computational grid. If a viscous layer is to be adequately defined, a fine grid is required in the vicinity of the blade surface and in such cases the stability limit would make an explicit calculation impractical. However, implicit methods are not subject to such stability limits, rather they are only limited by the physical time scale of the flow. Thus implicit solution procedures of the time-dependent Navier-Stokes equation represent a reliable method of solution for the blade passage problem. In addition, a time-dependent solution could be readily extended to investigate the time-dependent blade passage problem.

A typical time step in the Ref. 10 procedure consists of a time-wise linearization followed by a fully implicit difference approximation which is solved by an ADI (Alternating Direction Implicit) procedure of the Douglas-Gunn type (Ref. 12). The advantage with ADI methods is that a short sequence of simple matrix inversions replaces the complicated matrix inversion problem associated with a direct solution of the implicit equations. In this way a real savings in computer time is made without sacrificing accuracy or stability. By experience the scheme has run accurately and stably with time steps that are about 100 to 1,000 times the explicit stability limit. The net result is a solution procedure that for certain classes of problems is several orders of magnitude faster than the well-known explicit methods such as those of Lax-Wendroff (Ref. 11) and MacCormack (Ref. 13). The UTRC computer code MINT (Multidimensional, Implicit, Navier-Stokes, Time-dependent) which is based upon a time-dependent ADI solution of the Navier-Stokes equations, is a highly modularized efficient program which has options for two and three dimensional modes of operation. This code could form a basis for the efficient and accurate solution of the transonic viscous blade passage problem. Such a solution would include viscous effects, be extendable to three-dimensional and time-dependent problems and would not require specification of a Kutta condition.

Once the numerical method of solution has been chosen, the numerical solution of practical viscous flow fields is a function of the basic geometry of the flow region. For the overall numerical solution to be accurate and efficient, it is important to obtain an accurate representation of the geometry which is efficiently generated. In most practical problems of importance the flow region is generally nontrivial. Nontrivial geometries are usually treated by the use of either a finite element method or by a transformation of coordinates. In the finite element method, the numerical solution and the geometric analysis coincide (Ref. 14). Here small triangular or rectangular elements are used to cover the entire flow region; thus, forming a finite element mesh. The solution procedure results from an integration over each mesh element. In two dimensions the finite element analysis has been extended to include elements with one curved side. However, the analysis has not been effectively extended to three dimensions; and in addition, the two-dimensional case is not completely adequate for the accurate representation of nontrivial geometries. The problem occurs with the coupling of the finite element integration and its element boundaries which may be curved. Consequently the manner in which curved sides of a pair of finite elements are joined is quite restricted if the following integration is to be sufficiently simple. The restrictions imposed by integration tend to cause a poor representation of the geometric boundaries of the flow region. Here it is of prime importance to accurately represent the local curvature of the region boundaries. The boundary curvature can be accurately rendered if the boundary is fit without the above restriction. From this point of view, the use of a coordinate transformation is desirable. An additional attraction of a coordinate transformation approach is that a regular ordering of the computation mesh is given for a problem with curved boundaries. The regular mesh ordering is important since the matrix banded structure in the solution procedure is by far less complex than the typical structure which results from a direct use of the finite element method. The use of a coordinate transformation does not, however, rule out the possible use of a finite element method since the regular mesh in computational space can be used in a finite element solution of the transformed equations.

For the two-dimensional airfoil or cascade of airfoils, the most common type of coordinate transformation is the conformal transformation. The theory of one complex variable can be applied to obtain not only conformal coordinates but also a potential flow solution to the inviscid equations. The arbitrary airfoil, however, cannot be easily transformed via complex variables. The result is usually a composition of several conformal transformations which is generally complicated and time consuming. In addition, the potential flow solution is of little value in a study of the fully viscous flow field since the inviscid streamlines, as noted above, are not very accurate. Consequently, the use of a conformal transformation is not completely justified by the result of a potential flow solution. A further problem with conformal transformations is that highly cambered and/or tightly spaced airfoils are difficult to transform into reasonable coordinate curves. Often the positions of coordinate curves are impractically situated for a viscous calculation. The best positions would be expected from a

potential flow solution which can be obtained by the specification of a suitable source and sink, but then stagnation points occur at leading and trailing edges. The resultant coordinate singularities tend to cause difficult numerical problems. Such problems can be removed by abandoning the potential flow solution in attempting to obtain a suitable coordinate transformation. It is possible to consider alternate conformal mapping transformations not based upon potential flow solutions. Unfortunately the coordinate curves in such cases tend to have a spurious behavior. For example, one may obtain cascade coordinates with coordinate curves that connect upper and lower surfaces of the airfoil with s-shaped curves that pass around and over either the leading or trailing edges. A final drawback to the use of conformal transformations is that a straightforward extension to three dimensions does not exist. To date there has been no applicable function theory as in the two-dimensional case with complex variables. As a result the most common method of coordinate construction is to use a set of parallel planes each with a conformal transformation of the airfoil (or cascade) contours, (Jameson Ref. 15). As a further result, these nonorthogonal coordinates require a great amount of labor. In addition, the resulting coordinates are not orthogonal at the airfoil surface where large viscous gradients exist and must be suitably resolved.

All of the above problems are readily overcome with the cascade coordinate systems developed here. The coordinates are easily generated and well-positioned for the numerical study of a viscous flow field. The coordinates are nonorthogonal but this nonorthognoality is controlled. Specifically, the coordinates are orthogonal on the airfoil surface and gradually depart from orthogonality as the free stream regions are approached. Thus essentially orthogonal coordinates are used in the regions where large gradients appear in the solution. As orthogonality deteriorates the solution gradients become smaller; and hence, the projections between computational directions do not cause any significant errors in the numerical solution. The nonorthogonality does cause the equations of motion to contain more terms, but this is more than compensated for by the simple constructive nature of the cascade coordinates which result in an algorithm that is considerably faster than a corresponding conformal transformation. The only real time consuming part is the generation of an accurate surface representation which renders not only a good approximation to airfoil coordinate locations but also to the important airfoil curvature. A lesser demand on the accuracy of the surface representation would yield a faster algorithm; however, the accuracy of the final solutions would deteriorate. In addition, the cascade coordinate system presented herein allows a more flexible distribution of mesh points than does a conformal approach, and is easily extended to the three-dimensional case.

III. A COORDINATE SYSTEM FOR CASCADE GEOMETRIES

Overview

If a computational model of the viscous flow field through a cascade of airfoils is ever to become an effective design tool, then the resultant computer code must be of sufficient generality to accurately and efficiently produce physically reasonable solutions to a wide variety of cascade problems. To obtain the necessary generality use must be made of a coordinate system in which the cascade geometry is specified by a sufficiently dense discrete set of points. Ideally one may easily envision the design engineer who would enter his geometric data directly from his blueprints of an airfoil contour. In addition, other parameters must exist to allow the coordinates to be arbitrarily stretched in upstream and downstream directions so that free stream conditions can be well approximated. Also mesh distributions must be easily specified to adequately resolve regions where large gradients are to be expected, such as in the boundary layers and at leading and trailing edges.

The considerations indicated above have been well accounted for in the development of the cascade coordinates presented herein. The required input information consists of the geometric data points, the downstream-upstream extensions, the periodic spacing of airfoils, and the mesh point distribution. The periodic alignment of the cascade determines a vertical direction. The geometric data for the airfoil contour is then accepted in the form of a sequence of vertical slices of the airfoil. In properly aligned cartesian coordinates (x,y) each vertical slice consists of an x-coordinate which determines a vertical line $x = x_i$ and two y-coordinates $y = y_i$ and $y = z_i$ to denote the y values where the vertical line $x = x_i$ intersects the airfoil. (See Fig. 1).

It will be assumed that the distribution and quantity of vertical slices is enough to adequately specify the airfoil contour to a viewer who examines the airfoil as a whole from a reasonable distance. From a close up distance where only a small part of the airfoil is examined, the viewer would probably notice the occurrence of small fluctuations in the data caused by inaccuracies in measurement of this data. These inaccuracies will always occur when the data is taken directly from a graph of the airfoil contour. Data fluctuations of this type present no problem to the curve fitting procedures that are used here. The curve fitting will always be done with a parametric least-squares spline which will effectively filter out the unwanted noisy fluctuations in the data. The result is a smooth curve which accurately reflects the local curvature and the global shape of the given contour from which the discrete data were taken. In the development presented herein, it will be assumed that a least-squares spline routine is always available to convert discrete descriptions of curves into smooth and differentiable representations of the same curves.

Coordinate extensions in the upstream and downstream directions are specified by two input lengths which measure the distance of extensions, respectively, in front of and in back of the airfoil. The angles of the extensions are either input as parameters or are automatically made by the coordinate generation process. Thus

a user not only has the option to stretch his intended computational domain to obtain reasonable approximations of the far field regions of the flow; but also, he has the option to align the extensions with the flow direction into and out of the cascade. This will be seen to have the important property of concentrating the mesh points along the leading and trailing edges of the airfoil in an optimal fashion. An additional option on the distribution of mesh points is the control over the density of mesh points near the airfoil boundary. The user merely specifies a predicted boundary layer thickness and the fraction of mesh points that are desired to resolve the boundary layer. If desired, the predicted boundary layer thickness can easily be made a function of time by an updating process from the previous time levels in a solution procedure. The simplicity of this process will be evident from the modular way that this distribution enters into the coordinate construction. The remaining information that must be specified is the number of computational mesh points which are to be distributed around the outer boundary of the computational region. This number is broken down into three parts. Specifically, one must specify half the number of periodically aligned points, the number of points for the inflow boundary, and the number of points for the outflow boundary.

With the above input a system of coordinates is generated in a manner which is a generalization of both polar coordinates and the classical boundary layer coordinates (Ref. 16, pg 312). The circles in polar coordinates are now replaced by a family of loops about the airfoil which start with the airfoil itself and smoothly deform into the outer boundary of the computational region. The polar radii are replaced by straight lines which emanate from the airfoil surface and end on the outer boundary. As in boundary layer coordinates, the straight lines are taken to be orthogonal to the airfoil surface. However, since the intermediate loops are chosen by an interpolation between the airfoil surface and the outer loop the resultant system of coordinates is generally nonorthogonal. The nonorthogonality of the coordinate system as a whole is of no great concern since the coordinates are nearly orthogonal in the regions where the viscous flow is undergoing its greatest rate of change. Specifically, the coordinates, by construction, are precisely orthogonal along the airfoil surface and gradually deviate from orthogonality as one leaves the airfoil surface. The greatest degree of nonorthogonality occurs in the upstream and downstream regions where free stream conditions are being approached; and where, therefore, the gradients in the viscous flow field are very small. If the outer loop could be taken as a uniform expansion of the airfoil along its outward normal lines, then the resultant coordinate system would be precisely a set of boundary layer coordinates for the airfoil and accordingly would be orthogonal everywhere (Fig. 2). If an addition, the airfoil contour is a circle, then the coordinate system becomes a set of polar coordinates. Since the coordinate system has been constructed for a cascade of airfoils, the outer loop generally cannot be taken as an outward and uniform expansion of the airfoil itself (as in Fig. 2). Instead the outer loop must be generated from a curve which can conveniently be used for the application of the necessary periodicity conditions in the cascade problem. The

basic shape of this curve should be reasonably close to the camber of the airfoil. Hopefully, without too much confusion, it will be referred to as a camber curve. The generation of the camber curve is accomplished on a discrete level within the airfoil contour and is extended by lines outside of the airfoil. The discrete data can be conveniently generated by averaging the y-values y_i, z_i of each vertical slice $x = x_i$ in the discrete specification of the airfoil contour (depicted in Fig. 1). The discrete data is made into a differentiable curve which is extended by lines in front of and in back of the airfoil. Within the airfoil the camber curve is an approximation to the real camber line of the airfoil. The main distinction between these curves is that the real camber line is generated by averaging airfoil data along lines which are orthogonal to itself as opposed to the camber curve here which is generated by averages along the verticals. The camber curve thus has the advantage of a simple specification. The camber curve is shown in Fig. 3 as line A-B. After a smooth camber curve is created, the domain of the calculation is bounded by two curves each parallel to the camber curve. One curve, line C-D, is above the airfoil and the second curve, line E-F, is below the airfoil. The curves are separated by a distance equal to the airfoil spacing and are capped off at upstream and downstream ends by curves which are smoothly joined to form the differentiable outer loop depicted by CDFE in Fig. 3. The outer loop is then reparameterized in a manner which yields a periodic alignment for the mesh points where a periodicity condition must be applied. The next step is to impose the parameterization of the outer loop upon the inner loop by dropping normals onto the airfoil surface. The reparameterization is accomplished by the assignment of the parameter value of each outer loop point to the point on the airfoil contour which has an airfoil normal line passing through the given outer loop point. This is computationally executed on a discrete level and then made into a smooth curve by the least-squares spline routine. The inner loop with the imposed parameterization from the outer loop is now properly aligned so that any line joining inner and outer loop points of the same parametric value results in a line that leaves the airfoil as a normal line. The normal lines form the family of the coordinate curves which correspond to the radii of a polar system. These are illustrated in Fig. 3. The other coordinate curves consist of the loops that are obtained by an interpolation along the above normal lines. The periodic alignment of the resulting coordinate system is illustrated by the line GH which is represented by a dashed line since it is not a coordinate curve.

The Effect of Airfoil Curvature on the Placement of a Lower Coordinate Boundary

The construction of the cascade coordinate as outlined above will now be discussed in more detail. The necessary input data will be assumed to exist. The input data basically consists of a discrete rendition of the airfoil geometry by vertical slices, specified extensions of the computational domain upstream and downstream of the cascade, and the desired mesh point distributions for the flow field calculation.

As a first step the discrete airfoil data is converted into a smooth curve by an application of the least-squares spline algorithm. The curve parameterization is obtained from the cumulative arc length between data points. The result is an accurate fit to data with an (almost) arc length parameterized curve $\vec{\gamma}(t) = (\gamma^1(t), \gamma^2(t))$ which has at least two continuous derivatives and accurately reflects the curvature of the contour from which the raw data was taken. The two continuous derivatives are needed to perform the calculation of the curvature which is given by the formula

$$K = \frac{1}{\dot{S}^2} \sqrt{\ddot{\gamma}^m \ddot{\gamma}^m - \ddot{S}^2} \tag{1}$$

where S is the actual analytic arc length along the curve, $\dot{S} = \sqrt{\dot{\gamma}^m \dot{\gamma}^m}$, $\ddot{S} = \dot{\gamma}^m \ddot{\gamma}^m / \dot{S}$, and the summation convention of summing like indices has been invoked with $\dot{\gamma}^m \dot{\gamma}^m = \dot{\gamma}^1 \dot{\gamma}^1 + \dot{\gamma}^2 \dot{\gamma}^2$, etc., $\dot{\gamma}^m = d\gamma^m/dt$, and $\ddot{\gamma}^m = d^2\gamma^m/dt^2$. The analytic arc length, S, and the polygonal arc length, t, are nearly equal since t is an approximation of S. Thus $\dot{S} \simeq 1$, $\ddot{S} \simeq 0$, and as a result $K \simeq \sqrt{\ddot{\gamma}^m \ddot{\gamma}^m}$. The curvature here is needed to determine the extent of the computational boundary below the airfoil. Since the bottom side of the airfoil is usually concave it is clear that there is a restriction on the distance that the coordinates can extend below the airfoil. Otherwise, the proposed coordinate normal lines would have intersections among themselves when the domain is stretched beyond a certain point. This would cause coordinate singularities at such points. An illustration of this singularity is given in Fig. 4. To prevent the appearance of singularities due to intersecting normal lines, the cascade coordinates must be restricted in the region below the airfoil to lie above all points of possible intersections. The restriction is analytically specified by a knowledge of the centers of the oscillating spheres along the concave side of the airfoil. The osculating sphere in two dimensions is the circle which is tangent to the airfoil bottom and is determined by matching its derivatives with the airfoil surface until all of the parameters of the circle are determined. The result is a circle of radius 1/K which is tangent to the airfoil bottom. The center is located at a distance of 1/K along the airfoil unit normal vector $\hat{n}$ which is given by

$$\hat{n} = \frac{\ddot{\gamma}^m \dot{S} - \dot{\gamma}^m \ddot{S}}{K \dot{S}^3} \hat{u}_m \tag{2}$$

where $\hat{u}_1$ and $\hat{u}_2$ are the standard cartesian unit vectors along the x and y axes respectively. Thus, the vector position of the center is given by the quantity

$$\vec{\gamma} + \frac{\hat{n}}{K} \tag{3}$$

which will trace out a curve below the airfoil as the concave part of the airfoil bottom is traversed. To obtain a well-defined coordinate system the coordinates must terminate within the region bounded below by this curve and above by the airfoil bottom. For an illustration, see Fig. 5.

In the figure, the curve determined by the centers of the osculating spheres is given by the dotted curve, the lower boundary of a well-defined coordinate system is given by the dashed curve, and one of the osculating spheres is displayed as a solid curve. The curve which must be properly positioned below the airfoil is the curve which is parallel to the camber curve and which will be used to form the bottom of the coordinate system. This curve is chosen because it is bent in roughly the same manner as the airfoil. A good restriction on the vertical lowering of the camber curve is to lower it by a distance of not more than half way between the airfoil bottom and the curve determined by the centers of the osculating spheres. This restriction will place some distance between lower boundary of the coordinates and the centers of the oscillating spheres. The distance must be small enough to avoid potential problems which would result if the lower coordinate boundary were to approach the center of an osculating sphere. In such a case, small distances along the lower coordinate boundary would produce large corresponding distances along the airfoil bottom. The result would be an under resolution of the bottom of the airfoil and hence an undesirable loss of accuracy in a numerical solution for the flow field. For an illustration see Fig. 6. The rather uniform mesh distribution on the lower boundary is denoted by a sequence of x's and the correspondingly poor mesh distribution along the airfoil bottom is denoted by a sequence of dots. To obtain the desired lower bound on the amount that the camber curve can be lowered, half the minimum vertical distance from the bottom of the airfoil to the trace of center points of the oscillating spheres must be calculated. This is accomplished on a discrete level. The analytic curve for the airfoil is discretized by a uniform mesh over its parameterization and at each of these mesh points a unit normal vector (Eq. 2) is computed when possible. At inflection points the curvature vanishes and the unit normal vector given in Eq. 2 does not exist. Otherwise, the unit normals always exist and point in the direction of curve concavity. This is easily seen from Fig. 5 and the observation that the center of an oscillating sphere is in the positive normal direction. Consequently, a change in the direction of curve concavity can only occur at inflection points. A method is then needed to detect those inflection points where a change in the direction of concavity has occurred. The chosen mesh may not explicitly contain any of the inflection points but this is really no problem. The mesh is considered as an ordered set of points starting at airfoil leading edge and moving in a counterclockwise direction along the airfoil bottom, around the trailing edge, and then back along the top. At the leading edge, the normal vector is pointing into the interior of the airfoil and here an integer value of -1 is assigned. At the next point, the unit normal vector is projected upon the previous unit normal vector (which, in this case, is at the leading edge) by means of a dot product. If the data points are reasonably close together then the dot product is nearly +1 or -1. If it is nearly +1, then the

integer value is unchanged. Otherwise, a sign change is given. The process is then repeated and mesh points are successively assigned integer values of plus or minus unity. Specifically, the value of -1 is retained until the first mesh point in the concave portion of the underside of the airfoil is reached. From there the value of +1 is maintained until the next change in concavity occurs at the first mesh point where the airfoil becomes convex again which is usually near the trailing edge. If there is to be only one concave portion of the airfoil, the process can be terminated at this point since all other values would be -1 indicating that the remaining normals all point inwards. It is now clear that by ordering in a counterclockwise direction as opposed to the opposite direction, the indicated possibility of early termination occurs and is a way of conserving computational effort. (See Fig. 7.)

In concurrence with the above process, when points occur with assigned integer values of +1, the centers of the osculating spheres are calculated from the expression (Eq. 3). The x-coordinate of the center of an osculating sphere must usually lie within an interval determined by the x-coordinates of two successive airfoil mesh points. The only other intervals are the infinite intervals upstream and downstream of the airfoil. To find this interval a search is performed along the bottom of the airfoil. The integer locations are saved and the vertical distance can easily be computed as the average of the y-values at the interval endpoints minus the y-value of the center of the oscillating sphere. For an illustration see Fig. 8. The mesh along the bottom of the airfoil is denoted by a sequence of x's and the center of an oscillating sphere is given by a normal extending a distance of 1/K outside of the airfoil. As this process continues throughout the concave part of the airfoil bottom, the successive vertical distances are monitored and a minimum is taken. Next, the maximum vertical thickness of the airfoil is computed. With this information a criterion can easily be constructed to determine the amount which the camber curve is lowered to form the bottom of the computational domain. The difference between the periodic spacing and the maximum thickness of the airfoil is just the smallest vertical distance between consecutive airfoils as their chords are traversed. If one half of this distance is less than the allowable vertical distance due the concavity restriction above, then the bottom of the computational domain is set at one half of the periodic spacing distance below the camber curve. The top of the computational region is then one half of the periodic spacing distance above the airfoil; and the computational domain is bounded from above and below by well-centered curves which are parallel to the camber curve. By contrast if the inequality is in the other direction, then the camber curve is lowered by one half of the maximum airfoil thickness plus the distance due to the concavity restriction. This results in a computational domain which is not as well centered about the airfoil as in the previous case.

The Construction of the Camber Curve

The above criterion for the vertical displacements of the camber curve can all be generated before the camber curve is constructed. For its application the camber curve must obviously be in existence; and therefore, shall now be constructed. Since the airfoil data is specified as a sequence of vertical slices $x = x_i$ with y-values $y = y_i$ and $y = z_i$ (which are assumed to be upper and lower surface points respectively), airfoil camber data is generated by a criterion of the form $x = x_i$, $y = (1-\alpha_i)y_i + \alpha_i z_i$ for $0 < \alpha_i < 1$, as i runs through all of the vertical slices. For a smooth set of camber data the sequence α_i must be generated from a continuous function of limited total variation (see Royden, Ref. 17). The result of any such choice of function will be a sequence of data points which roughly follow the camber of the airfoil since all data points must lie within the interior of the airfoil. For convenience, the sequence $\alpha_i = 1/2$ was selected. The resulting sequence of data points is first parameterized by polygonal arc length and then fit with a least-squares spline curve. The fit will vary in accuracy depending upon the number of spline segments that are used. Generally the accuracy will increase with the number of segments when everything else remains unchanged. In this case, however, accuracy is less important than it was with the airfoil contour. Consequently, a smaller number of segments are needed. Under the assumption that the curve remains reasonably near the data, the only additional constraint on accuracy is that the polygonal arc length parameterization provides a reasonable approximation to the analytic arc length of the resultant curve. This part of the camber curve is used to form upper and lower computational boundaries which are directly over and under the airfoil itself. On these parts of the outer boundary loop it is important to obtain a reasonably uniform mesh distribution with respect to arc length since it is this mesh distribution which will be used to impose a parameterization over most of the airfoil surface as normal lines are dropped. A uniform subdivision of the parameter values will then result in a uniform distribution of mesh points over the segments in question on both the airfoil and the surrounding outer loops. An additional bonus is that the linear extensions of the camber curve in upstream and downstream directions are simplified. Since the parameter is almost an arc length parameter, it has an arc length derivative which is nearly unity. On the linear extensions a continuous rate of expansion relative to arc length is desired so that the number of mesh points are conserved as the computational boundaries are sketched. Otherwise the numerical computation of the viscous flow field would overly resolve the stretched regions, and thus waste a considerable amount of computational time on parts of the flow where no substantial changes are occurring. Typical extensions in the upstream and downstream directions would be on the order of one chordal length of the airfoil in each direction. The number of potentially wasted points then could be substantial.

Since the linear expansion is to occur smoothly from an existing arc length parameterization, the arc length derivative of the parameter must be unity where the extensions are joined to the curve. The direction of each extension is given by the specification of a unit vector in the desired direction. Typical choices of

direction may, for example, be selected from the flow conditions or from the global airfoil geometry. More specifically, one may stretch the coordinate system in the free stream directions of the far field velocity vectors upstream and downstream of the cascade. Or one may stretch the upstream and downstream extensions in the direction of the real airfoil camber line as it emerges from each end of the airfoil. This latter choice is a result of the global geometry since the direction of the real camber line is bent by the airfoil contour under the requirement that it be the midpoint of its own normal line which intersects the airfoil on opposite sides. When the vertically averaged airfoil data is fit with a small number of splined segments, the accuracy of the fit is lowered and the alignment with the global geometry is improved. At the leading and trailing edges of the airfoil the unit tangent vectors to the camber curve are well aligned with the global geometry and are nearly in the same direction as the endpoint tangent vectors of the real camber line. By contrast, when a large number of splined segments are used, a highly accurate fit is obtained and the unit tangent vectors at the endpoints of the resultant curve are in directions which generally only reflect the local airfoil geometry and not the airfoil camber. For example, if the leading and trailing edges of the airfoil are formed by circular arcs, then vertical averages over circles will result in lines parallel to the x-axis (which is generally not aligned with the endpoint camber directions) and this may or may not be a desirable direction for an extension of the coordinate system. Thus, the directions upon which the camber curve leaves the airfoil are controlled by the number of splined segments which are used in the construction of the curve. When the directions of extension are not specified, the camber curve will be extended in the directions which it leaves the airfoil and the directional control, as seen above, will be a result of the number of splined segments.

The directions in question are obtained from the unit tangent vectors to the camber curve at leading and trailing edges. Let $\vec{\alpha}(t)$ for $0 \le t \le t_1$ denote the camber curve between leading and trailing edges. Then the vector field consisting of unit tangent vectors to $\vec{\alpha}(t)$ is given by

$$\vec{v}(t) = \frac{d\vec{\alpha}}{dS} \simeq \frac{d\vec{\alpha}}{dt} \tag{4}$$

where S is the arc length starting from $\vec{\alpha}(0)$. The approximate equality is a result of the polygonal approximation of t to S. Since the vector field $\vec{v}$ points in the direction of increasing arc length along $\vec{\alpha}$, the extension in front of the leading edge is in the negative $\vec{v}(0)$ direction. Thus, an extension in front of length F is of the form

$$\vec{\alpha}(0) + S_F(t)\ \vec{v}(0) \tag{5}$$

where S_F is the arc length measured from -F to 0 as the parameter t varies from 0 to some T. The value of T will determine the proportionate number of points in front of $\vec{\alpha}$ relative to the number of points on $\vec{\alpha}$. If $\vec{\alpha}(t)$ is discretized into k points uniformly distributed with respect to t, then the parameter spacing is given by $\Delta t = t_1/(k-1)$. For the extension in front, the greatest integer part of $F/\Delta t$ (denoted $[F/\Delta t]$) is a measure of the number of whole Δt increments that could be fit into the extension if the parameter t of the extension were to approximate arc

length. For small extensions it is desirable to continue the parametric approximation to arc length by a discretization of the extension into the number of parametric intervals just given. However, if the extension is large a coordinate stretching relative to arc length is best. Large extensions are often needed to approximate free stream conditions. Thus, the extension is cut into

$$n_F = \min([F/\Delta t], m_F) \tag{6}$$

units of length Δt where the positive integer m_F is a specified cut off value. The arc length function $S_F(t)$ is then to be parameterized from 0 to $T = n_F \Delta t$. At the value T the derivative of S_F is taken to be unity since the extension is to be joined at the resultant point with the nearly arc length parameterized curve $\vec{\alpha}$. The desired stretching is readily given by the quadratic arc length function

$$S_F(t) = (T-t)[(\frac{1}{T} - \frac{F}{T^2})(T-t) - 1] \tag{7}$$

which monotonically increases from $S_F(0) = -F$ to $S_F(t) = 0$ and ends with a slope of $S_F'(T) = 1$. A graph of S_F appears in Fig. 9. If F is large the function leaves -F fairly rapidly and gradually decreases its rate of climb towards 0 where the rate of increase becomes proportional to arc length. Upon substitution into the expression for the linear extension in front (5), a discretization for $t = 0, \Delta t, \ldots, n_F \Delta t$ yields data points on the line which at the start are separated by fairly large distances that continually decrease until the end where the separation is proportional to arc length. When the parameters of the discretization of $\vec{\alpha}$ are each increased by the addition of T units, the result is a discretization of the extension in front continued by the discretization of $\vec{\alpha}$ with the new parameterization which varies smoothly through the juncture point. Note that the juncture point is produced by both curves but is only counted once in this process. Thus, the discretization consists of $k+n_F$ points counting endpoints. The last point has a parameter value of $t_2 = t_1 + T$, and a rate of change that is directly proportional to arc length. In the same manner as before, an extension of B units in length is added on to form a linear continuation in back of the airfoil trailing edge. The extension is of the form

$$\vec{\alpha}(t_1) + S_B(t)\vec{v}(t_1) \tag{8}$$

where S_B is the arc length measured from t_2 to $t_2 + B$, as the parameter varies from t_2 to $t_2 + R$ for some length R. The value of t_2 was chosen instead of t_1 since the eventual discretization will be a continuation of the previous discretizations, and this choice will immediately lead to a smooth continuation of the parameterization. The extension in back is cut into

$$n_B = \min([B/\Delta t], m_B) \tag{9}$$

units of length Δt with an integer cut off value of m_B for a total parameter change of $R = n_B \Delta t$. Unlike the extension in front, however, the extension in back expands from an arc length parameterization and not into one. This change in direction results in the requirement that $S_B'(t_2) = 1$. As before, a quadratic stretching function is sufficient and the result is given by

$$S_B(t) = (t - t_2)[(\frac{B}{R^2} - \frac{1}{R})(t - t_2) + 1] \tag{10}$$

which monotonically increases from $S_B(t_2) = 0$ to $S_B(t_2 + R) = B$ and starts with a slope $S_B'(t_2) = 1$. A graph appears in Fig. 10 .

The discretization for $t = t_2 + \Delta t \ldots, t + n_B \Delta t$ yields data points on the linear extension in back which at the start are separated by distances that are proportional to arc length and then continually increase from there in an opposite fashion to the frontal extension. When this discretization is added onto the end of the prior discretization, a properly parameterized discretization of the entire camber curve with extensions is obtained. The parameterization starts from 0 and ends with a value $t_3 = t_2 + R$. The total number of points in this discretization is given by the sum $n_F + k + n_B$. At this stage the data points could be fit with smooth curve that is parameterized in correspondence with the given discrete parameterization. Instead, however, it is best to directly use the above discretization to form a properly parameterized discretization of the entire outer loop which encircles the airfoil and forms the outer boundary of the coordinate system. Then the outer loop discretization will be fit with just one curve fitting process as opposed to separate curve fits which must be smoothly joined between the upstream and downstream endcaps and the vertically translated camber curves. For an illustration, see Fig. 11 where the constituent parts of the outer loop have been displayed. The camber curve appears as the curve which linearly emerges from the airfoil through its extensions. The vertical translates are then displayed along with the adjoining endcaps. At the expense of a small amount of storage the discretization of the camber curve is vertically translated above and below the airfoil in the manner prescribed by the algorithm of the previous section which yields a determination of the vertical distance of translation based upon the airfoil thickness, spacing, and underside curvature.

The Construction of Endcaps for the Outer Loop

Let the vertically translated camber curves of the previous section be denoted by $\vec{p}(t)$ and $\vec{q}(t)$ for the lower and upper parts of the outer loop. As with the camber curve itself, the range of the parameter t will be from 0 to t_3 as the curves are traversed from front to back. To complete the specification of the outer loop, endcaps must be constructed to join the translated camber curves together, at both the front and back ends. For the construction of endcaps, it is sufficient to use

a bicubic curve at each end with the stipulation that function values and tangent vectors are matched at the joins. It will be assumed that the parameter values are taken from 0 to some number T. It is often convenient to set T equal to the periodic spacing distance. However, this choice is arbitrary. The adjustment to a larger value of T will only cause the bicubic to bulge out further than before and, in this regard, the bulge can be used to stretch the coordinates in a similar manner to the earlier extensions of the camber line. Unlike the extensions of the camber line, the extensions due to this bulging action have no periodic alignment and only tend to separate points where upstream and downstream data must be specified. Thus, it is much more desirable to stretch the coordinates with only the camber line extensions and not such bulges. Each cubic polynomial $a_o + a_1 t + a_2 t^2 + a_3 t^3$ is determined by a system of the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} e_o \\ e_1 \\ e_2 \\ -e_2 \end{bmatrix} \tag{11}$$

where e_o and e_1 are polynomial values at the respective endpoints o and T, and e_2 is the slope at o. The system is easily solved to yield the polynomial

$$e_o + e_2 t + \left[3\left(\frac{e_1 - e_o}{T^2}\right) - \frac{e_2}{T}\right] t^2 - 2\left(\frac{e_1 - e_o}{T^3}\right) t^3 \tag{12}$$

For the x-coordinates, the endpoint evaluations are equal due to the vertical alignment of the camber lines. Consequently, $e_o = e_1$ and the polynomial becomes a quadratic which starts with a slope of e_2 and ends with a slope of $-e_2$, (see Fig. 12). In the front $e_o = p_1(o)$ and in the back $e_o = p_1(t_3)$ where the decomposition $\vec{p} = (p_1, p_2)$ has been used. The slope e_2 is given by the x-component of the direction of camber line extension which is $-\vec{V}(o)$ for the front and $\vec{V}(t_1)$ for the back where $\vec{V}(t)$ is given by Eq. (4). The negative slope of $-e_2$ is needed at T since the parameter values are increasing in a direction opposite to the unit vector which points in the direction of camber line extensions. The y-components of these slope conditions are used to evaluate the quantity e_2 for the calculation of the y-coordinates. The polynomial evaluations for the y-coordinates are given by $e_o = p_2(o)$, $e_1 = q_2(o)$ for the front and $e_o = q_2(t_3)$, $e_1 = p_2(t_3)$ for the back. Note that the orientation is from bottom to top in front and from top to bottom in back. This is done as a matter of convenience so that a clockwise parameterization can be easily given to the outer loop. The graph of one of the cubic y-coordinates is given in Fig. 13. The endcaps are then discretized by a sufficiently fine mesh

and parameterized by polygonal arc length. When taken with the translated camber lines the result is a discretized version of the entire outer loop. But one is still not finished since the parameters must be suitably adjusted to interface with the desired mesh point specification for the fluid dynamic calculation. Suppose that the computational mesh is to have k periodic points along the translated camber lines p(t) and q(t), n points along the front endcap not counting juncture points, and m points along the back endcap also not counting juncture points. The total computational mesh along the outer loop would then consist of n+2k+m points; and therefore, that same number of normal lines to the airfoil surface. The parameterization of the translated camber lines $\vec{p}(t)$ and $\vec{q}(t)$ is the same and varies from 0 to t_3. Since these parameterizations will be preserved up to rigid translations, the interval 0 to t_3 is cut into a uniform mesh with k-1 intervals of length $\Delta t = t_3/(k-1)$. At the front endcap n+1 such intervals are needed. Thus, the arc length parameterization of the front endcap must be replaced by a parameterization from 0 to $(n+1)\Delta t$ which smoothly passes through the juncture points. This is accomplished by a blend of straight lines from each endpoint with slope determined by the existing arc length derivative $\dot{S}$ at those points. This process is illustrated in Fig. 14.

The lines are generated with arc length S as the independent variable which varies from 0 to S*, the arc length of the endcap on the front. The slope of each line is given by the rate at which the camber curve parameter varies with arc length at the beginning of the camber curve. Thus, one has the two parallel lines

$$\ell_o(S) = \frac{S}{S_F'(o)} \tag{13a}$$

and

$$\ell_1(S) = \frac{S-S^*}{S_F'(o)} + (n+1)\Delta t \tag{13b}$$

where $S_F(t)$ is given by Eq. (7). The desired blend must start along ℓ_o, gradually leave ℓ_o, and smoothly merge into ℓ_1 to end at $(S^*, (n+1)\Delta t)$. This is accomplished with a linear homotopy (Ref. 18) between ℓ_o and ℓ_1 with a homotopy deforming parameter given by the function

$$h(S) = \frac{f(S) - f(o)}{f(S^*) - f(o)} \tag{14a}$$

where

$$f(S) = \tfrac{1}{2}\left[1 + \tanh\left\{\frac{2D}{S^*}\left(S - \frac{S^*}{2}\right)\right\}\right] \tag{14b}$$

with a damping factor D which controls the rate of ascention between the lines. Usually a value of D which lies between 2 and 3 is quite satisfactory. A graph of h is given in Fig. 15. The resultant parameterization is then given by the linear homotopy

$$t(S) = [1-h(S)]\, \ell_o(S) + h(S)\, \ell_1(S) \qquad (15)$$

which is used to reparameterize the front endcap. Then the parameterization along the upper curve $\vec{q}(t)$ is shifted by the addition of $(n+1)\Delta t$ to each parameter value. The result is a discretized curve with a smooth parameterization covering the front endcap with parameter values from 0 to $(n+1)\Delta t$ and continuing along the top with $(k-1)\Delta t$ units to end at a parameter value of $(n+k)\Delta t$. At this stage, the endcap at the back is adjoined, and in the same manner as above, it is reparameterized to vary smoothly from $(n+k)\Delta t$ to $(n+k+m+1)\Delta t$ where a juncture occurs with the lower curve $\vec{p}(t)$. Then the orientation of the lower curve is reversed by the relabeling of points so that one has the curve $\vec{p}((k-1)\Delta t-t)$. The resulting parameterization for $\vec{p}$ is next, shifted by $(n+k+m+1)\Delta t$ units so that a smooth parameterization is properly specified for the entire outer loop. The outer loop is given by a discrete set of points which are parameterized from 0 to $(n+2k+m)\Delta t$ as the loop is traversed in a clockwise direction. If desired, one can renormalize the parameterization so that it varies from 0 to 1. The result of a renormalization is only a rescaling of the parameterization. At this stage, an application of the least-squares spline procedure is used to transform the outer loop data into a smooth curve with three continuous derivatives and the prescribed parameterization. Note that the least-squares procedure will effectively filter out the small smoothness errors that occurred when arc length was approximated with the arc length of a polygonal curve. The small smoothness errors in question appeared as slope information at the juncture points on each end of the camber curve extensions. Parametric accuracy within the camber curve is not very important since the periodic alignment of mesh points is not affected by a slight loss of accuracy within that region. On the endcaps such questions of accuracy would seem more important. However, the construction above was performed in a manner where the accuracy did not enter into the assignment of parameter values at the endpoints of any endcap. The only effect then would be in the specification of slopes for the lines ℓ_o and ℓ_1 (for each endcap) which would undergo the smoothing of least squares anyhow. Consequently, the alignment of periodic points will be very accurate.

The Reparameterization of the Airfoil Surface

Now that the outer loop of the coordinate system is constructed with a suitable parameterization, one must reparameterize the airfoil surface to align airfoil parameter values with outer loop parameter values so that corresponding points lie on the same airfoil normal line. Once the reparameterization has been accomplished, the coordinate transformation will be given by the cartesian equation

$$\vec{x} = R(y^2)\vec{\alpha}(y^1) + [1 - R(y^2)]\,\vec{\beta}(y^1) \tag{16}$$

where $\vec{x} = (x^1,x^2)$ are cartesian coordinates, $\vec{\beta}$ is the airfoil contour, $\vec{\alpha}$ is the outer loop, R is a coordinate distribution function along the normals, and $\vec{y} = (y^1,y^2)$ are curvilinear coordinates. The coordinate, y^2, is the position along normal lines, and y^1 is the position around the outer loop which is to be imposed upon the airfoil surface and hence upon all intermediate coordinate loops. The reparameterization is accomplished in a discrete manner. A sufficiently dense uniform mesh is used to discretize the outer loop parameter; and hence, to create a smooth sequence of outer loop points with their smoothed parameter values. From each of these points a normal line must be dropped to the surface of the airfoil. The simplest way to find the desired airfoil normal line is to locate the point on the airfoil surface which is closest to the outer loop point in question. For each outer loop point, this distance minimization problem is always solvable since the airfoil surface is either locally convex or is locally concave with the centers of the oscillating spheres removed a sufficient distance beyond the outer loop. This latter result occurred by construction when a determination was made on the amount to lower the camber line to form the lower boundary of the coordinate system. When the airfoil point of minimum distance is located, it is assigned the parameter value of the outer point. The process is then continued to the next point on the outer loop until all data points on the outer loop have been used. The result is a discrete reparameterization of the airfoil surface which can be turned into a smooth curve in either of two ways. First, the airfoil may be recreated by treating the given airfoil data as raw data and directly applying the curve fitting routine. If the reparameterization should cause enough distortion relative to arc length, then it is best to consider a curve fit to the change of parameter relationship. That is, the second method is to pair off new and old parameter values in the above process and then to fit the resulting curve. The reparameterized airfoil is given by the composition with the old parameterization expressed as a function of the new parameterization. Consequently, the airfoil geometry remains invariant in this process and the accuracy and rendition of airfoil curvature is preserved. Thus, when the original fit to the airfoil is a good one, the second method is desirable.

An integral part of the reparameterization is the method used to drop the normals. For reasons of simplicity and stability, the algorithm is based on the minimization of distance, as indicated above. The outer loop mesh points are consecutively taken in the clockwise ordering of the parameterization starting with the lower juncture point of the front endcap and the lower camber curve. At the first point, the distance to an airfoil data point within the leading edge region is computed. Then a search is performed over the existing airfoil data with cartesian x-coordinates less than the above distance. This criterion limits the search to a region around the leading edge. The result is a fairly rapid determination of the existing data point of minimum distance. For an illustration see Fig. 16 where d is the distance to the leading edge region. An arc of radius d and centered at the first outer loop point $\vec{\beta}(0)$ is used to determine the vertical line tangent to the arc which appears as the dashed line $x = \beta_1(0) + d$. This vertical cuts the airfoil into two parts. To the right of the line the points on the airfoil must be greater than d and hence need not be considered. Thus, the search is performed on the smaller region to the left which contains the leading edge. The location of the airfoil data point of minimum distance is then used to start the search algorithm used on the remaining points. The algorithm starts with a known previous position. For the first point, this position is assumed to be the location determined above. For other outer loop points, the previous position is taken to be the existing airfoil data point just before the point on the airfoil determined by the normal line dropped from the previous outer loop point. Since outer loop points are taken in a clockwise order, the previous point on the airfoil is simply the existing airfoil data point which is nearest to the point in question when distance is measured only in the couterclockwise direction. From here a distance is computed and the search over existing data is continued until the measured distance exceeds the starting distance. This process limits the search of existing data points to a small region on the surface of the airfoil, and thus, saves computer time. For an illustration see Fig. 17. The mesh on the outer loop is denoted by a sequence of dots and the existing airfoil data is denoted by a sequence of x's. The distance, d_1, to the previous airfoil data point is measured along a line (dashed in the figure) which generally intersects the previous normal line unless the normal line emanates precisely from an existing airfoil data point. The search is continued until one reaches a distance, d_2, which is greater than the starting distance, d_1. In the illustration, the search would result in the selection of the point of minimum distance from the first three points pictured. After the distance has been minimized over the existing airfoil data, the analytic formulation of the airfoil contour is used to create new data for a refined search in a small neighborhood of the point determined from the search of existing data. The simplest procedure is to search for a point of minimum distance over a smaller discretization around the locality in question. A uniform mesh is placed on the corresponding parameter values so that a dense enough discretization is obtained between the nearest existing airfoil data points on either side of the minimum point. If the first point in the search of existing data is the point of minimum distance, then the mesh refinement need only cover the

interval from that point to the nearest point in the clockwise direction. This smaller interval can be used to increase accuracy or the speed of computation. In Fig. 17 the first point is displayed as the point of distance d_1. A move in the counterclockwise direction would only increase the distance; and thus, not contribute to the search for minimum distance. For more accuracy further local refinements could be taken to form a nesting of refinements. However, when the accuracy is of the same order as the curve fitting accuracy, there is little need to continue the search to greater perfection. The local search presented here is probably the crudest of all known techniques. At the expense of extra programming logic more efficient techniques can be applied. One of the easiest methods to apply is the method of Hooke and Jeeves'. The search in that method is broken up into a sequence of exploratory and pattern moves. For details on this and other methods see the text by J. R. Walsh (Ref. 19). However, for the one dimensional airfoil surface considered in this application, the payoff of a more efficient optimization technique is negligible and, in fact, is probably less efficient when the additional logic has been added. On the other hand, if the natural three dimensional extension of the coordiante construction presented herein is to be done, then the method of search is more important and a more efficient optimization technique should be used.

Distribution Functions

When partial differential equations are discretized in terms of differences, the derivatives are replaced in some fashion by difference quotients. A simplification then leads to the difference equations that we solve. Implicitly in the discretization, however, is the assumption that derivatives are accurately estimated by secant lines. But then the exact solution may experience drastic variations in a short distance. Such solutions are said to have large gradients. In regions where the gradients are large, the approximation of derivatives by secants may be very poor unless the particular region is disected into smaller regions which have reasonable secant approximations, a practice commonly known as mesh refinement. In fluid mechanics, the boundary layer of a viscous flow around or through an object is such a region.

Obviously, the necessary resolution could be accomplished by merely increasing the number of points in a uniform distribution; however, this would require excessive computer time and storage. Another alternative, known as the interface method, is to use a refined mesh only in the given region and then join it with the global mesh. An improved technique is to use coordinate distribution functions which smoothly distribute mesh points so that in some sense they are spaced in roughly an inverse proportion to the size of the gradients. Thus, regions of high gradients have proportionately more points than regions with smaller gradients. Unlike the interface method, the transition between different mesh lengths is made continuously, and as gradually as possible. Distributions are often used when the distributional transformation is applied to an independent variable of an existing transformation.

The result is a new transformation obtained by composition. With this approach, the problem of mesh point distribution is replaced by the problem of selecting a suitable set of distribution functions within a transformation of coordinates. The problem is a nontrivial one since the distribution functions should depend upon the nature of the solution being computed but are determined in advance of the computation. Thus, some prior knowledge of the solution is required. In flows with large boundary layer separation or with adjacent dissimilar components, the critical region to be resolved is somewhere in the middle of the flow. But the location of such regions is often unknown at the outset of the problem. One method to overcome this difficulty in marching procedures is to create the distribution function at the next level based upon a knowledge of the solution at the present level. Care must be taken, however, to create a distribution function that is sufficiently smooth in the marching direction. In many problems of practical interest, however, the regions that need resolution are known in advance. Typical examples are attached boundary layers and boundary layers that may have small separations or separation bubbles.

Within the framework of cascade coordinate systems boundary layer resolution on the inner surface is accomplished by setting

$$R(y^2) = 1 + \left(\frac{a}{b} y^2 - 1 \right) \tanh \left[D \left(\frac{1-y^2}{1-b} \right) \right] \tag{17}$$

where a is the estimated boundary layer thickness, b is the desired proportion of mesh points in the boundary layer, and D is the hyperbolic dampling factor. The boundary layer growth a gives the fraction of the flow region occupied by the boundary layer, b is usually taken as a constant, and D can be given a value of about 2. When y^2 is small, the radial distribution of equation (17) reduces essentially to the line

$$R \simeq \frac{a}{b} y^2 \tag{18}$$

which would have been chosen had we used the interface method. As y^2 approaches unity the distribution Eq. (17) smoothly approaches unity as illustrated in Fig. 18.

IV. THE GENERATION OF THE NAVIER-STOKES EQUATIONS WITH THE METRIC DATA FOR CASCADE COORDINATE

The efficient generation of metric data is an important part of any solution procedure involving general curvilinear coordinates. Before a solution can be undertaken, the physical problem must be specified. Problem specification, however, involves the creation of boundary and initial data and the generation of the equations of motion with the associated boundary conditions. In addition, the solution may be monitored, examined, or put under physical constraints. In all of these tasks, the metric data is needed. A knowledge of the metric data is enough to completely specify the equations of motion and analyze the coordinate invariant directions for the specification of boundary and initial conditions. For very complicated geometries the equations of motion may contain an inordinate number of terms. However, if the equations are taken in tensor form, then the coefficients to terms can be constructed from the metric data with the construction process being performed on a computer. Once a nontrivial term is constructed, its contribution to the desired difference equations is computed before searching for the next nontrivial term. Sequentially, the process continues until all terms in the equations have given their contributions to the system of difference equations. Then, in the same fashion, one cycles through terms in the boundary conditions, sequentially adding in their respective contributions. The result is the desired set of difference equations, and the problem is effectively reduced to linear algebra. Note that with such methods there is no real need to write out the differential equations or complicated boundary conditions in detail. Thus, all one needs to do is to generate the metric data and use it.

The coordinate transformation from cascade coordinates into cartesian coordinates is given by Eq. (16) in the previous section. By differentiation of the coordinate transformation, one obtains the Jacobian transformation which leads directly to the transformation rules for tensor fields. These rules allow one to input, monitor, or extract basic information from a solution procedure involving transformed variables. The Jacobian Transformation is essentially obtained from the chain rule which yields

$$\vec{e}_i = \frac{\partial \vec{x}}{\partial y^i} = \frac{\partial x^j}{\partial y^i} \frac{\partial \vec{x}}{\partial x^j} = \frac{\partial x^j}{\partial y^i} \hat{u}_j \qquad (19)$$

where $\hat{u}_j$ is the standard orthonormal basis of constant vector fields, and $\vec{e}_j$ is the natural basis of tangent vectors to coordinate curves. With a slight abuse of notation, $\vec{x}$ has been used as a position vector in the definitions of $\vec{e}_j$ and $\hat{u}_i$. However, nothing is lost since the covariant derivative of $\vec{x} = x^j\hat{u}_j$ is just the partial derivative of the x^j summed on $\hat{u}_j$. In terms of the notation

$$\hat{u}_1 = \begin{pmatrix} 1 \\ o \end{pmatrix} \quad \text{and} \quad \hat{u}_2 = \begin{pmatrix} o \\ 1 \end{pmatrix} \tag{20}$$

one has

$$\vec{e}_i = \begin{pmatrix} \frac{\partial x^1}{\partial y^i} \\ \frac{\partial x^2}{\partial y^i} \end{pmatrix} \tag{21}$$

and hence the Jacobian matrix

$$(\vec{e}_1, \vec{e}_2) = \begin{pmatrix} \frac{\partial x^1}{\partial y^1} & \frac{\partial x^1}{\partial y^2} \\ \frac{\partial x^2}{\partial y^1} & \frac{\partial x^2}{\partial y^2} \end{pmatrix} \tag{22}$$

In the standard cartesian basis $\hat{u}_i$ the outer loop and the airfoil contour are expressed in the form $\vec{\alpha} = \alpha^i\hat{u}_i$ and $\vec{\beta} = \beta^i\hat{u}_i$, respectively. In this notation, the transformation for cascade geometries is given in component form by the equations

$$x^i = R(\alpha^i - \beta^i) + \beta^i \tag{23}$$

for $i = 1,2$. By differentiation the Jacobian transformation is given by

$$\vec{e}_1 = \left[R\left(\frac{d\alpha^i}{dy^1} - \frac{d\beta^i}{dy^1}\right) + \frac{d\beta^i}{dy^1}\right]\hat{u}_i \tag{24a}$$

and

$$\vec{e}_2 = \left[\frac{dR}{dy^2}(\alpha^i - \beta^i)\right]\hat{u}_i \tag{24b}$$

The metric tensor g_{ij} is obtained from the differential element of arc length $(ds)^2 = g_{ij}\, dy^i dy^j$. From the known cartesian form and an application of the chain rule, the differential element of arc length is expanded through the sequence of equalities

$$(ds)^2 = d\vec{x}.d\vec{x} = (\frac{\partial \vec{x}}{\partial y^i}\, dy^i).(\frac{\partial \vec{x}}{\partial y^j}\, dy^j) = \frac{\partial \vec{x}}{\partial y^i} \cdot \frac{\partial \vec{x}}{\partial y^j}\, dy^i dy^j = (\vec{e}_i \cdot \vec{e}_j)\, dy^i dy^j \quad (25)$$

and, as a result, the metric is given by the equation

$$g_{ij} = \vec{e}_i \cdot \vec{e}_j \quad (26)$$

The $\vec{e}_j$-direction covariant derivative D_j of the vector $\vec{e}_i$ is again a vector and hence is expressible in terms of the same basis $\vec{e}_1$, $\vec{e}_2$. Specifically,

$$D_i \vec{e}_j = \Gamma_{ij}^m \vec{e}_m \quad (27)$$

where the coefficients Γ_{ij}^m are known as Christoffel symbols. This covariant derivative measures the rate of change of $\vec{e}_j$ along a coordinate curve in the direction of $\vec{e}_i$. This coordinate curve is an integral curve of $\vec{e}_i$ which is obtained by fixing all except the i^{th} variable in the transformation.

The assumption will be made that the covariant derivative is the natural one derivable from the metric. This is known as the Levi-Civita connection (Ref. 20). The Christoffel symbols for this covariant derivative are given by the formula

$$\Gamma_{ij}^k = \frac{g^{km}}{2} \left\{ \frac{\partial g_{mj}}{\partial y^i} + \frac{\partial g_{mi}}{\partial y^j} - \frac{\partial g_{ij}}{\partial y^m} \right\} \quad (28)$$

where the g^{km} are elements of the matrix inverse to the matrix of metrics (g_{ij}). This formula is easily obtained by differentiating $g_{ij} = \vec{e}_i \cdot \vec{e}_j$ with respect to y^m, permuting all three of these indices, forming the sum in parenthesis, applying symmetry to the lower indices of the Christoffel symbols, and then applying the inverse metric. With some calculation, one can obtain the nonzero Christoffel symbols directly from the above formula.

For the automatic computation of the metric data it is convenient to use forms which are explicitly given in terms of the coordinate transformation and its derivatives. By a direct expansion of the dot product in Eq. (26) the components of the metric tensor become

$$g_{ij} = \frac{\partial x^\ell}{\partial y^i} \frac{\partial x^\ell}{\partial y^j} \tag{29}$$

If A denotes the Jacobian matrix of Eq. (22), then it is easy to see that the matrix (g_{ij}) is given by

$$g = \det(g_{ij}) = \det\,(A^t A) = \det(A^t)\,\det(A) = (\det A)^2 = J^2 \tag{30}$$

where J is used to denote the Jacobian of the transformation. For nonsingular transformations J is nonzero and hence both A and (g_{ij}) are invertable. Thus, the inverse metric is obtained from

$$(g^{km}) = (g_{ij})^{-1} = (A^t A)^{-1} = A^{-1}(A^t)^{-1} = A^{-1}(A^{-1})^t \tag{31a}$$

which is converted into components to yield

$$g^{km} = \frac{\partial y^k}{\partial x^\ell} \frac{\partial y^m}{\partial x^\ell} \tag{31b}$$

The Christoffel symbols can now be obtained by a direct substitution into Eq. (28). This yields the expansion

$$\Gamma^k_{ij} = \frac{1}{2} \frac{\partial y^k}{\partial x^r} \frac{\partial y^m}{\partial x^r} \left\{ \frac{\partial^2 x^\ell}{\partial y^i \partial y^m} \frac{\partial x^\ell}{\partial y^j} + \frac{\partial x^\ell}{\partial y^m} \frac{\partial^2 x^\ell}{\partial y^i \partial y^j} + \frac{\partial^2 x^\ell}{\partial y^j \partial y^m} \frac{\partial x^\ell}{\partial y^i} + \frac{\partial x^\ell}{\partial y^m} \frac{\partial^2 x^\ell}{\partial y^j \partial y^i} - \frac{\partial^2 x^\ell}{\partial y^m \partial y^i} \frac{\partial x^\ell}{\partial y^j} - \frac{\partial x^\ell}{\partial y^i} \frac{\partial^2 x^\ell}{\partial y^m \partial y^j} \right\} \tag{32}$$

which by cross cancellation collapses into the form

$$\Gamma_{ij}^{k} = \frac{\partial y^{k}}{\partial x^{r}} \frac{\partial y^{m}}{\partial x^{r}} \left\{ \frac{\partial x^{\ell}}{\partial y^{m}} \frac{\partial^{2} x^{\ell}}{\partial y^{i} \partial y^{j}} \right\} \tag{33}$$

But the inner two factors are just the product of the Jacobian transformation and its inverse. Consequently, they may be replaced by the Kronecker symbol δ_{r}^{ℓ} which is unity if $r = \ell$ and vanishes otherwise. On substitution, the Christoffel symbols are now given by the simple expression

$$\Gamma_{ij}^{k} = \frac{\partial y^{k}}{\partial x^{\ell}} \frac{\partial^{2} x^{\ell}}{\partial y^{i} \partial y^{j}} \tag{34}$$

which is suitable for automatic computation.

In terms of arbitrary metric data, the governing equations are derived from the Navier-Stokes equations for the compressible flow of a viscous, perfect gas. The resulting expressions are given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x^{i}} (\rho u^{i} \sqrt{g}) = 0 \tag{35}$$

for continuity and

$$\left[\frac{\partial}{\partial t} (\rho u^{k} \sqrt{g}) + \frac{\partial \sigma^{jk}}{\partial y^{j}} + \sigma^{ij} \Gamma_{ij}^{k} \right] \vec{e}_{k} = 0 \tag{36}$$

for momentum where

$$\sigma^{ij} = (\rho u^{i} u^{j} + \tau^{ij}) \sqrt{g} \quad . \tag{37}$$

and τ^{ij} are the components of the stress tensor in the tensor product basis $\vec{e}_i \otimes \vec{e}_j$. Constant total temperature is assumed, and thus an energy equation is not required. The primitive solution variables above are given by a specification of the stress tensor which in expanded form is given by

$$\tau^{ij} = g^{ij}\, p + a_k{}^{ij}\, v^k + b_k{}^{i\ell j}\, \frac{\partial v^k}{\partial y^\ell} \tag{38a}$$

where

$$a_k{}^{ij} = \mu \left(\frac{2}{3}\, g^{ij}\, \Gamma_{k\ell}^{\ell} + \frac{\partial g^{ij}}{\partial y^k} \right) \tag{38b}$$

and

$$b_k{}^{i\ell j} = \mu \left(\frac{2}{3}\, g^{ij}\, \delta_k^{\ell} - g^{i\ell}\, \delta_k^{j} - g^{j\ell}\, \delta_k^{i} \right) \tag{38c}$$

for viscosity μ and Kronecker deltas $\delta_j^i = \delta^{ij} = \delta_{ij}$.

From the ideal gas law and the constant total temperature assumption, the perfect gas relation has the form

$$p = A\rho + B\rho g_{ij} v^i v^j \tag{39}$$

where A and B are constants.

If desired, the momentum equation can easily be put into conservation law form. When the expression for the Christoffel symbols given in Eq. (34) is inserted into the momentum equation (36), one obtains

$$\left[\frac{\partial}{\partial t} \left(\rho v^k \sqrt{g} \right) + \frac{\partial \sigma^{jk}}{\partial y^j} + \sigma^{ij}\, \frac{\partial y^k}{\partial x^\ell}\, \frac{\partial^2 x^\ell}{\partial y^i \partial y^j} \right] \vec{e}_k = 0 \tag{40}$$

A change of basis from the curvilinear direction $\vec{e}_k$ into the cartesian directions $\hat{u}_m$, can be expected to simplify the momentum equation. This is performed by an application of Eq. (19) which yields

$$\left[\frac{\partial}{\partial t} \left(\rho v^k \sqrt{g}\, \frac{\partial x^m}{\partial y^k} \right) + \frac{\partial x^m}{\partial y^k}\, \frac{\partial \sigma^{jk}}{\partial y^j} + \sigma^{ij}\, \delta_\ell^m\, \frac{\partial^2 x^\ell}{\partial y^i \partial y^j} \right] \hat{u}_m = 0 \tag{41}$$

With the assumption of nonmoving coordinates the Jacobian transformation was brought through the time derivative. Now the definition of the Kronecker symbol is applied and the dummy indices i in the last term are replaced by k's. The result is given by

$$\left[\frac{\partial}{\partial t}\left(\rho U^{k}\sqrt{g}\,\frac{\partial x^{m}}{\partial y^{k}}\right)+\frac{\partial x^{m}}{\partial y^{k}}\,\frac{\partial \sigma^{jk}}{\partial y^{j}}+\sigma^{jk}\,\frac{\partial}{\partial y^{j}}\left(\frac{\partial x^{m}}{\partial y^{k}}\right)\right]\hat{u}_{m}=0 \qquad (42)$$

which, in component form, reduces to the system of conservation laws

$$\frac{\partial}{\partial t}\left(\rho U^{k}\sqrt{g}\,\frac{\partial x^{m}}{\partial y^{k}}\right)+\frac{\partial}{\partial y^{j}}\left(\sigma^{jk}\,\frac{\partial x^{m}}{\partial y^{k}}\right)=0 \qquad (43)$$

For more information on this topic see Refs. 21 and 22

Although the rather formal develpment above provides a specification of a problem in the cascade coordinate system, it does not provide much insight into the metric structure which is needed to interpret results and to properly apply boundary conditions. For this reason the metric will be derived in terms of the basic geometric parameters of the cascade. Once this is done, correlations between the metric structure and the underlying coordinates can be made. It is first observed that the cascade coordinate transformation (16) can be broken down into two basic parts. Since $\vec{\alpha} - \vec{\beta}$ is a nontrivial normal vector pointing from the airfoil $\vec{\beta}$ to the outer loop $\vec{\alpha}$, its magnitude $d = \|\vec{\alpha} - \vec{\beta}\|$ is a measure of the distance across the coordinate system in the direction given by the outward unit normal vector from the airfoil. However, the outward unit normal is given by both

$$\hat{n} = \frac{\vec{\alpha} - \vec{\beta}}{\|\vec{\alpha} - \vec{\beta}\|} \qquad (44)$$

and the Frenet formulas on the airfoil contour. For the airfoil contour a unit tangent vector $\hat{\tau}$ is given by $\hat{\tau} = \dot{S}\, D_1 \vec{\beta}$ where $\dot{S}$ is the y^1 - derivative of arc length along the airfoil. Upon successive differentiation one obtains the Frenet formulas

$$\begin{aligned} D_1 \hat{\tau} &= c\,\dot{S}\,K\,\hat{n} \\ D_1 \hat{n} &= -c\,\dot{S}\,K\,\hat{\tau} \end{aligned} \qquad (45)$$

where $c = -1$ on convex parts of the airfoil and $c = 1$ on concave parts. At inflection points, however, the formulas do not exist. For a derivation of the formulas one may consult a text on diffefential geometry (Ref. 20). Consequently, the coordinate transformation can be written in the form

$$\vec{x} = R\, d\, \hat{n} + \vec{\beta} \tag{46}$$

with the unit normal vector given by either of the above specifications. At non-inflection points the latter specification shall be used so that the Frenet formulas can be employed to some advantage. Since the coordinate transformation is constructed from functions each of only one variable, derivatives of these functions can be denoted with a dot and result in no ambiguities. In this notation, the transformation (46) is differentiated to obtain the natural basis of tangent vectors to coordinate curves. From an application of the Frenet formulas the result becomes

$$\begin{aligned} \vec{e}_1 &= R\dot{d}\,\hat{n} + \dot{S}\,(1 - cKRd)\,\hat{\tau} \\ \vec{e}_2 &= \dot{R}\,d\,\hat{n} \end{aligned} \tag{47}$$

For an illustration of the vector relationships see Fig. 19. Since the vectors $\hat{\tau}$ and $\hat{n}$ are orthonormal, the metric is readily obtained from a direct substitution into the equation $g_{ij} = \vec{e}_i \cdot \vec{e}_j$. The result is given by

$$\begin{aligned} g_{11} &= (R\dot{d})^2 + [\dot{S}\,(1-cKRd)]^2 \\ g_{12} &= g_{21} = R\,\dot{R}\,d\,\dot{d} \\ g_{22} &= (\dot{R}\,d)^2 \end{aligned} \tag{48}$$

and from the determinant of this metric one obtains the Jacobian

$$J = \sqrt{g} = d\,\dot{R}\,\dot{S}\,(1-cKRd). \tag{49}$$

The magnitude of the Jacobian, however, is a measure of the relative scaling of coordinate volume elements throughout the domain of the transformation. If the Jacobian is zero at a point, then the differential volume element there is zero and the transformation is singular. Since the Jacobian is a continuous function, one may also examine the coordinates as a singularity is approached. With the cascade coordinates presented herein, a singularity can occur only if one of the factors in the expression of the Jacobian should vanish. However, each of these possibilities will lead to an unreasonable system of coordinates. The factors $\dot{R}$ and $\dot{S}$ can be

eliminated from consideration since both R and S must be given by strictly monotone functions; and therefore, cannot vanish. A lack of monotonicity here would cause the coordinates to locally double back upon themselves; and thus, render local regions where the coordinates are not uniquely defined. This leaves one with two possible factors that could vanish. First, if d should vanish, then the airfoil surface and the outer loop would coincide at the point or points in question. As the points of coincidence are approached, the coordinate loops are then smoothly compressed into a region of zero cross section. An illustration of this type of singularity is given in Fig. 20. The second possibility for a sigularity would occur if the last factor (1-cKRd) should vanish. This, however, could only occur in the region of a concave part of the airfoil since otherwise the factor is the sum of positive quantities. But in the region of a concave part of the airfoil, the centers of the osculating spheres were sent outside of the coordinate system by construction. The analytic implication is that $Kd < 1$ and hence the factor cannot vanish.

The rate of change along coordinate curves is measured by the covariant derivatives of the natural coordinate tangent vectors. In this regard, the Christoffel symbols contain the desired information. For example, an application of the covariant derivative D_2 to $\vec{e}_2$ yields

$$D_2 \vec{e}_2 = \ddot{R}\, d\, \hat{n} = \frac{\ddot{R}}{\dot{R}}\, \vec{e}_2 \tag{50}$$

and hence the Christoffel symbols

$$\Gamma^1_{22} = 0 \quad \text{and} \quad \Gamma^2_{22} = \frac{\ddot{R}}{\dot{R}} \tag{51}$$

by observation from equation (27). This result is also partially evident from the basic geometry. The curves of constant y^1 are just the normal lines; and hence, any variation of their tangent vectors must be in magnitude only. This conclusion is born out from the analytic fact that Γ^1_{22} vanishes. In the special case of a uniform distribution of loops, the function R is given by $R = y^2$. The second derivative vanishes with the result that Γ^2_{22} also vanishes. Then $D_2\vec{e}_2 = 0$ which implies that $\vec{e}_2$ is constant along its normal line. Another example is given by coordinates with a region where d is constant. In that region, the natural tangent vectors to coordinate curves are given by

$$\begin{aligned} \vec{e}_1 &= \dot{S}\,(1\text{-}cKRd)\,\hat{\tau} \\ \vec{e}_2 &= \dot{R}\, d\, \hat{n} \end{aligned} \tag{52}$$

which are clearly orthogonal. The covariant derivative D_2 of $\vec{e}_1$ is given by

$$D_2\vec{e}_1 = -\dot{cKRd}\,\hat{\tau} = \frac{-\dot{cKRd}}{\dot{S}(1-cKRd)}\,\vec{e}_1 \tag{53}$$

and hence the Christoffel symbols

$$\Gamma^1_{21} = \Gamma^1_{12} = \frac{-\dot{cKRd}}{\dot{S}(1-cKRd)} \quad \text{and} \quad \Gamma^2_{21} = \Gamma^2_{12} = 0 \tag{54}$$

are obtained from Eq. (27), as before.

The result is again geometrically reasonable since the loopwise coordinate tangent vectors must all be parallel along a normal line. If in addition, the region were to contain the effect of a linear segment embedded in the airfoil surface, then the Christoffel symbols Γ^1_{12} and Γ^1_{21} would vanish and $D_2\vec{e}_1 = 0$. The coordinates would then be locally cartesian. Further, calculations and interpretations of the nature presented here can be done for the remaining Christoffel symbols.

RESULTS

To evaluate the algorithm for the generation of cascade coordinate systems described in the previous sections, a suitable test case was devised. The chief criterion was to obtain a test case which was complicated enough to simulate a real cascade, and yet specialized enough so that comparisons could be made with known geometric parameters. Since most real cascades are known to be composed of highly cambered airfoils, it was required that the test case would be for a cascade with a highly bent airfoil. In addition, since the cascade coordinates are generated from raw data, the test case was constrained to a problem where the airfoil curvature was known. In this way the geometric representation of the airfoil could be evaluated for accuracy in both location and curvature. Since circles are curves with known constant curvature, it was reasonable to construct the airfoil in the test case with circular arcs. Then with the exception of transitional regions near the junctures between consecutive arcs, the curvature could be compared with curvature of the underlying arc. With the above criteria, the airfoil was constructed from two concentric arcs of slightly different radii which were closed by smaller circular arcs attached to each end. An illustration is given in Fig. 21.

The two concentric arcs were constructed with an inner radius R_1 and an outer radius R_2. The center point was taken at $x_4 = 0$ and the arcs extended through angles from $\pi/4$ to $3\pi/4$ radians. To form a closed loop smaller arcs of radius $r = (R_2 - R_1)/2$ were attached to either end. These arcs were centered at the cartesian locations $(\pm x, x)$ with $x = (R_1 + r)/\sqrt{2}$. To express the data in terms of vertical slices the airfoil was subdivided into five regions where a unique analytic description was available. The regions are marked off by the dashed vertical lines in the figure. At either end the verticals through the intervals $[x_2, x_1]$ and $[x_7, x_6)$ cut the airfoil contour on only the small circular arcs. At the next inner most intervals $[x_3, x_2)$ and $[x_6, x_5)$ the bottom part of the airfoil contour is given by the small circular arcs and the top is given by the outermost circular arc of radius R_2. The centered interval $[x_5, x_3)$ leads to verticals which cut only the two concentric circular arcs of radii R_1 and R_2. The locations of the interval endpoints are readily determined to be

$$\begin{aligned} x_7 &= r + A & x_1 &= -x_7 \\ x_6 &= \frac{R_2}{\sqrt{2}} & x_2 &= -x_6 \\ x_5 &= \frac{R_1}{\sqrt{2}} & x_3 &= -x_5 \end{aligned} \qquad (55)$$

where $A = (R + r)/\sqrt{2}$.

On the small endcap arcs a local specification of angular position is given. This is illustrated by the angle α in the figure. Positions along the concentric arcs of radii R_1 and R_2 are given by the angular positions θ_1 and θ_2 respectively. The only constraint is to vertically align the data. For notational convenience let $u(x)$ denote the upper surface and $\ell(x)$ denote the lower surface. Then from the figure one has

$$\begin{aligned} x &= -A + r\cos\alpha \\ \ell &= A - r\sin\alpha \\ u &= A + r\sin\alpha \end{aligned} \tag{56a}$$

on $[x_2, x_1]$ for $\pi \le \alpha \le \frac{5\pi}{4}$,

$$\begin{aligned} x &= -A + r\cos\alpha \\ \ell &= A - r\sin\alpha \\ \theta_2 &= \cos^{-1}\{[-A + r\cos\alpha]/R_2\} \\ u &= R_2\sin\theta_2 \end{aligned} \tag{56b}$$

on $[x_3, x_2)$ for $\frac{5\pi}{4} < \alpha \le \frac{7\pi}{4}$,

$$\begin{aligned} x &= R_1\cos\theta_1 \\ \ell &= R_1\sin\theta_1 \\ u &= R_2\sin\{\cos^{-1}(\frac{R_1}{R_2}\cos\theta_1)\} \end{aligned} \tag{56c}$$

on $[x_5, x_3)$ for $\frac{\pi}{4} < \theta_1 < \frac{3\pi}{4}$, and a similar treatment for the remaining intervals. When the endcap angles α and the angle θ_1 for the inner concentric arc are discretized by a uniform mesh the result is a collection of vertical slices which discretely define the airfoil contour. For the test case the central interval $x_3 < x \le x_5$ was partitioned by 29 verticals determined by $\theta_1 = \frac{3\pi}{4} - \Delta\theta_1, \frac{3\pi}{4} - 2\Delta\theta_1, \ldots, \frac{3\pi}{4} - 29\Delta\theta_1 = \frac{\pi}{4}$ where $\Delta\theta_1 = \pi/60$. The other intervals $x_\ell < x \le x_{\ell+1}$ for $i = 1, 2, 5, 6$ were similarly partitioned with 9 verticals apiece resulting from subdivisions of the angles α on either endcap. In addition, a vertical slice was added at x_1. For simplicity in the interpretation of results, the inner radius R_1 was given a value of unity. This gave the concave part of the airfoil a curvature of unity. Again, for reasons of simplicity, the outer radius was given the value of 1.2 so that the

radius of the smaller endcap arcs would be r = .1 and thus the curvature there would be exactly 10. Coordinate stretches were specified by setting camber curve extensions of .5 in the upstream direction and .7 in the downstream direction. The inner part of the camber curve was, for simplicity, given as vertically averaged data with a specification for a very accurate fit. Next, the periodic spacing of airfoils was given a value of 3/4. The scaling of the coordinates should then be roughly 3 units across and 1 unit high. Consequently, on the average absolute errors should be about twice as large as relative errors. The number of computational mesh points on the outer loop was set by choosing 20 periodic points above and below the airfoil and 5 points on both the upstream and downstream endcaps. The radial distribution was set for a boundary layer region to occupy one quarter of the distance from the airfoil to the outer loop and to be resolved with one half of the mesh points. For aesthetic reasons, 7 radial points were chosen so that there would be 5 inner coordinate loops. The computation time for the calculation was slightly less than 30 seconds on a UNIVAC 1110. This compares favorably with other methods of computation and is, in fact, faster than most. Since the purpose of the present study was to obtain an accurate construction of cascade coordinate systems, little attention was actually paid to computational efficiency in terms of computer time. Consequently, with a little effort the computer time could be decreased even further. A graph of the results appears in Fig. 22. The airfoil contour was fit with a maximum absolute error of 4×10^{-3} in the location of points. The curvature along the concentric arcs were generally accurate to within two or three digits while the larger curvature regions on the leading and trailing edges were accurate to within only one or two digits. As expected the camber data was accurately fit with a maximum absolute error of 2×10^{-3}. As a result the linear data on the ends of the circular caps caused the camber curve extensions to leave the airfoil as straight lines parallel to the x-axis. The periodic alignment for periodically matched points was generally accurate to three decimal places and in some places had even greater accuracy. Certainly such excellent results cannot be visually discerned from the graph itself. However, it can be observed that the lines from the airfoil to the outer loop are for all practical purposes normal to the airfoil and again the result is in excellent agreement with the theory. Also the radial distribution, as expected, properly distributed the 5 inner loops.

REFERENCES

1. Yoshihara, H: Some Recent Developments in Planar Inviscid Transonic Airfoil Theory. AGARD-AG-156, 1972.

2. Delaney, R. A. and P. Kavanagh: Transonic Flow Analysis in Axial-Flow Turbomachinery Cascades by a Time-Dependent Method of Characteristics. ASME Paper No. 75-GT-8, 1975.

3. Gopalakrishna, S. and R. Bozzola: A Numerical Technique for the Calculation of Transonic Flows in Turbomachinary Cascades. ASME Paper No. 71-GT-42, 1971.

4. Garabedian, P. R. and D. G. Korn: Analysis of Transonic Airfoils. Communications on Pure and Applied Mathematics, Vol. XXIV, 1971, pp. 841-851.

5. Steger, J. L. and H. Lomax: Numerical Calculation of Transonic Flow About Two-Dimensional Airfoils by Relaxation Procedures. AIAA 4th Fluid and Plasma Dynamics Conference, Paper No. 71-569, 1971.

6. Murman, E. M. and J. D. Cole: Calculation of Plane Steady Transonic Flows. AIAA Journal, Vol. 9, No. 1, 1970.

7. Grossman, B. and G. Moretti: Time-Dependent Computation of Transonic Flows. AIAA 7th Annual Meeting and Technical Display, Paper No. 70-1322, 1970.

8. Erdos, J., P. Baronti, and S. Elzweig: Transonic Viscous Flow Around Lifting Two-Dimensional Airfoils. AIAA 5th Fluid and Plasma Dynamics Conference, AIAA No. 72-678, 1972.

9. Rehyner, T. and I. Flugge-Letz: The Interaction of Shock Waves with a Laminar Boundary Layer, International Journal of Nonlinear Mechanics, Vol. 3, 1968.

10. Briley, W. R. and H. McDonald: An Implicit Numerical Method for the Multidimensional Compressible Navier-Stokes Equations. United Aircraft Research Laboratories Report M911363-6, November 1973.

11. Richtmeyer, R. D. and K. W. Morton: Difference Methods for Initial Value Problems. J. Wiley & Sons, 1967.

12. Douglas, J. and J. E. Gunn: A General Formulation of Alternating Direction Methods. Numerische Math., Vol. 6, 1964, pp. 428-453.

13. MacCormack, R. W.: The Effect of Viscosity in Hypervelocity Impact Cratering. AIAA 7th Aerospace Sciences Meeting, Paper No. 69-354, 1969.

REFERENCES (CONT'D)

14. Fix, G. and G. Strang: An Analysis of the Finite Element Method. Prentice Hall, Inc., Engelwood Cliffs, New Jersey, 1975.

15. Jameson, A.: Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1, AIAA Conf. 1973.

16. Rouse, H., et al: Advanced Mechanics of Fluids, Wiley, New York, 1959.

17. Royden, H. L.: Real Analysis. MacMillan, New York, 1963.

18. Hu, S. T.: Homotopy Theory. Academic Press, Inc., New York, 1959.

19. Walsh, J. R.: Methods of Optimization. Wiley, New York, 1975.

20. Laugwitz, D.: Differential and Riemannian Geometry. Academic Press, Inc., New York, 1965.

21. Eiseman, P. R.: The Numerical Solution of the Fluid Dynamical Equations in Curvilinear Coordinates. AFWL-TR-73-172, August 1973.

22. Eiseman, P. R. and A. P. Stone: A Note On A Differential Concomitant. Proceedings of the American Mathematical Society, Vol. 53, No. 1, November 1975.

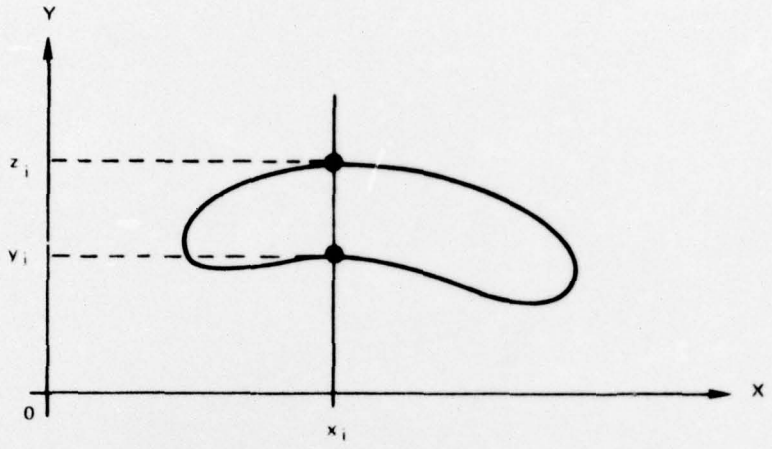

FIGURE 1: INPUT DATA FOR THE AIRFOIL CONTOUR

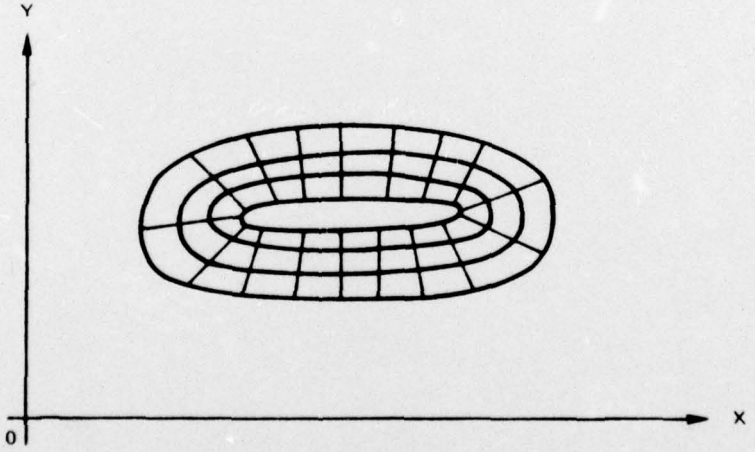

FIGURE 2: BOUNDARY LAYER COORDINATES

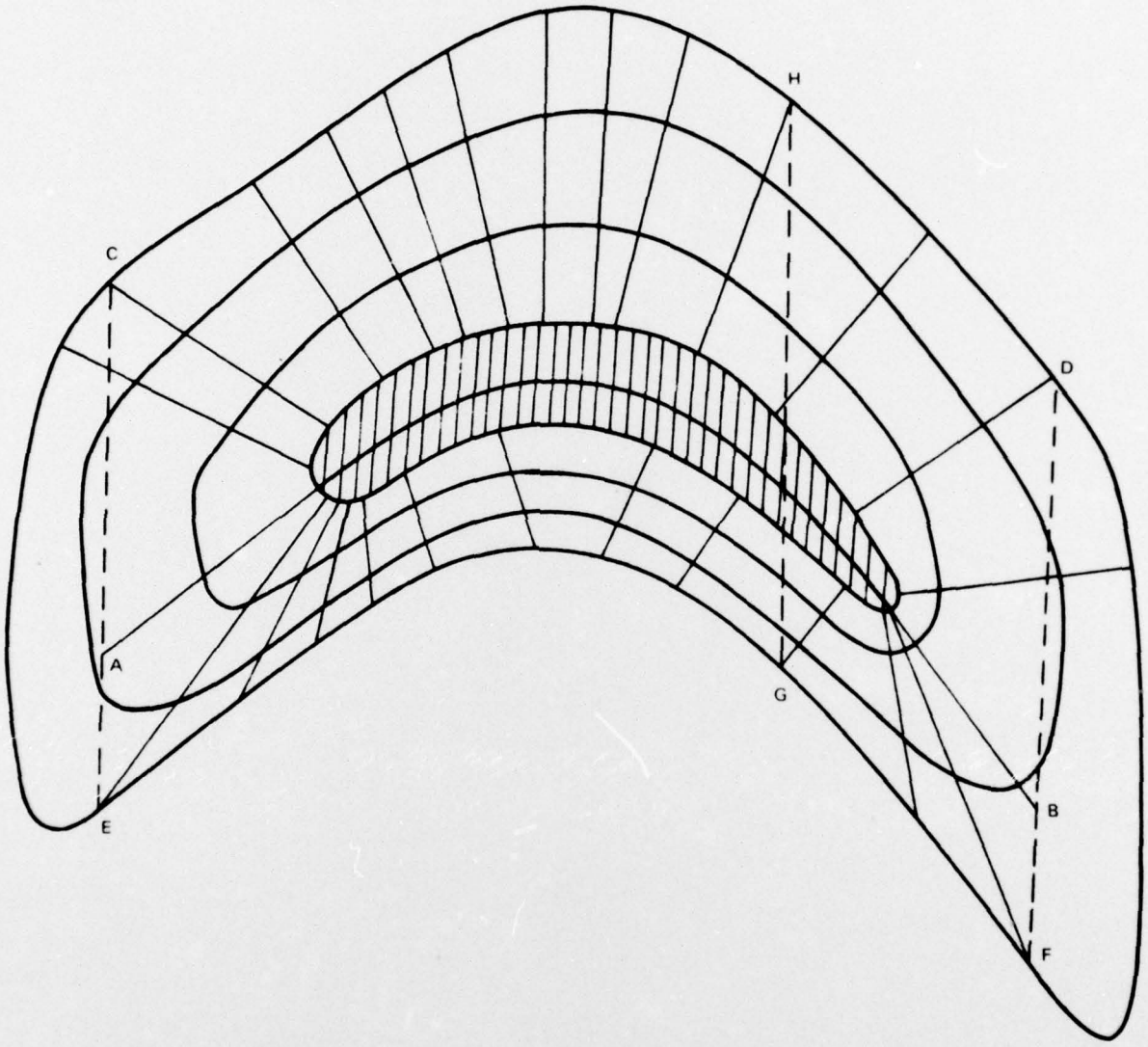

FIGURE 3: SCHEMATIC OF AIRFOIL COORDINATE SYSTEM

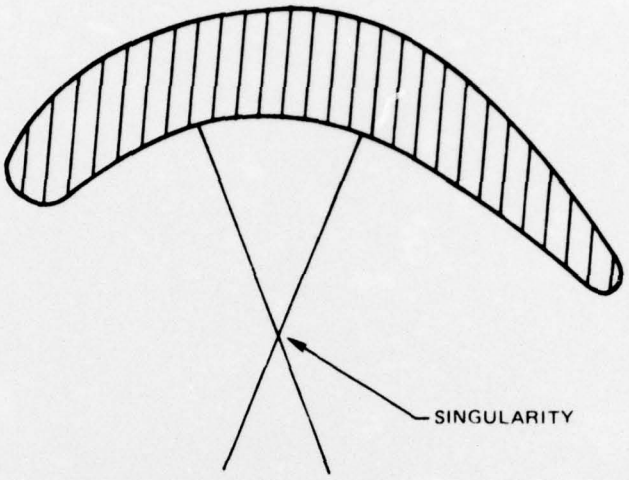

FIGURE 4: SINGULARITY FROM INTERSECTING NORMALS

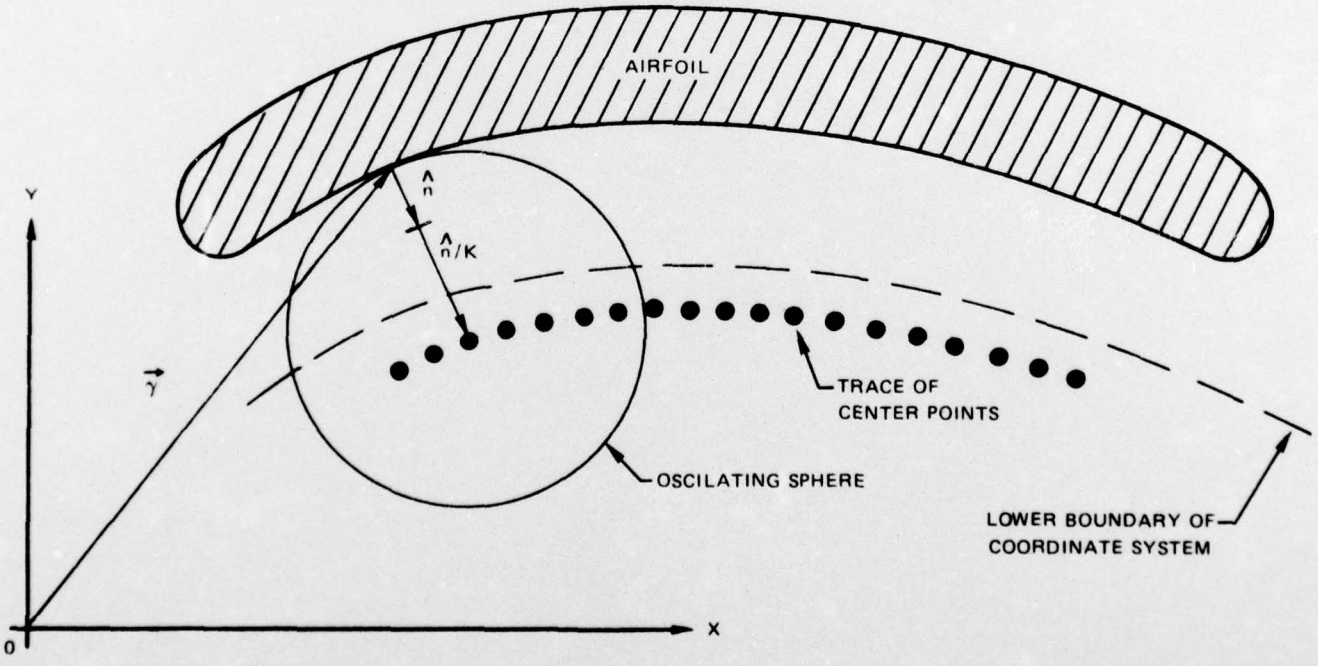

FIGURE 5: DETERMINATION OF THE LOWER COORDINATE BOUNDARY

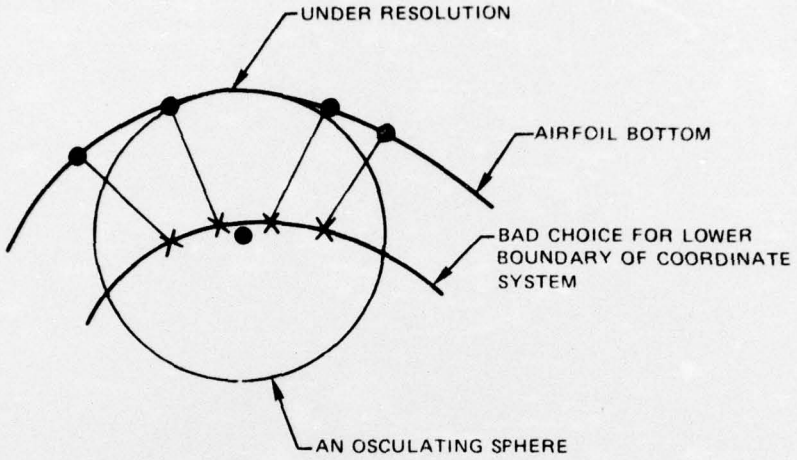

FIGURE 6: A COORDINATE SYSTEM WHICH IS NEAR THE CENTER OF AN OSCULATING SPHERE

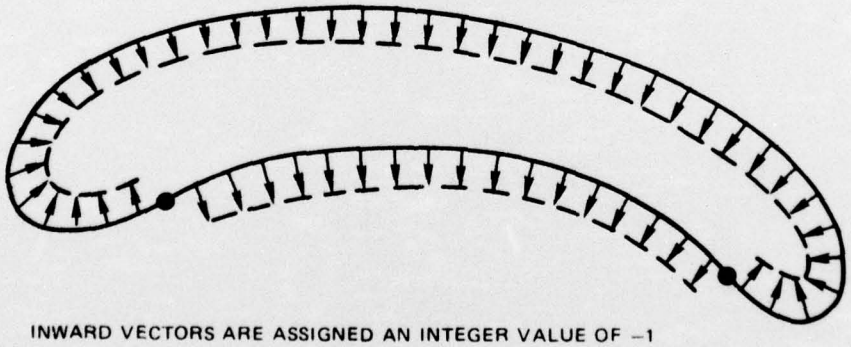

FIGURE 7. DISCRETE NORMAL VECTOR FIELD ALONG AIRFOIL SURFACE

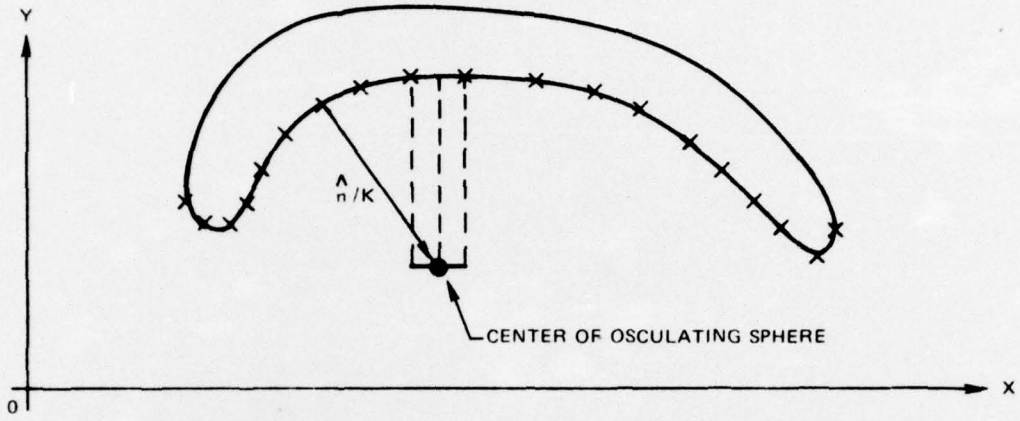

FIGURE 8: THE VERTICAL DISTANCE FROM THE BOTTOM OF THE AIRFOIL TO THE CENTERS OF THE OSCULATING SPHERES

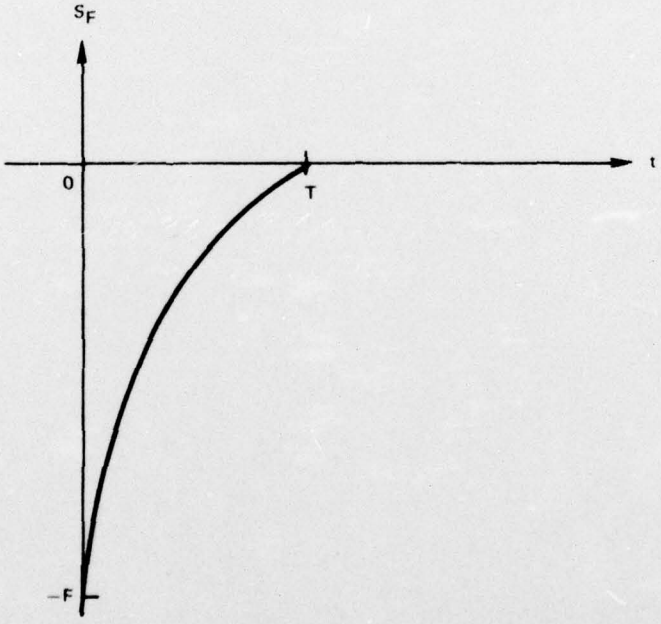

FIGURE 9: ARC LENGTH FUNCTION FOR EXTENSION IN FRONT

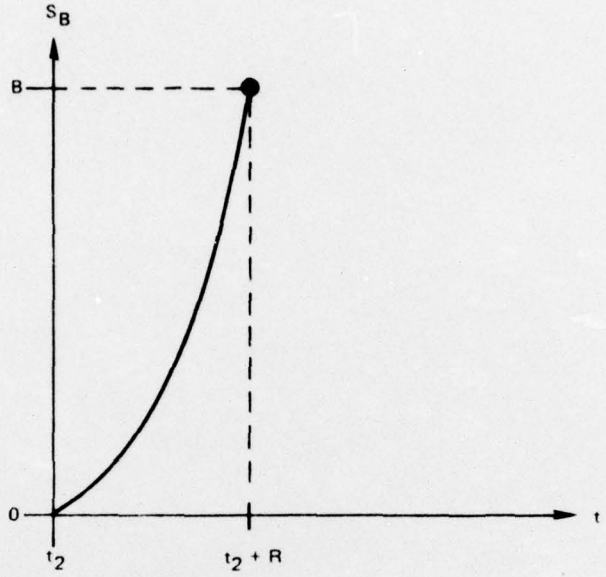

FIGURE 10: ARC LENGTH FUNCTION FOR EXTENSION IN BACK

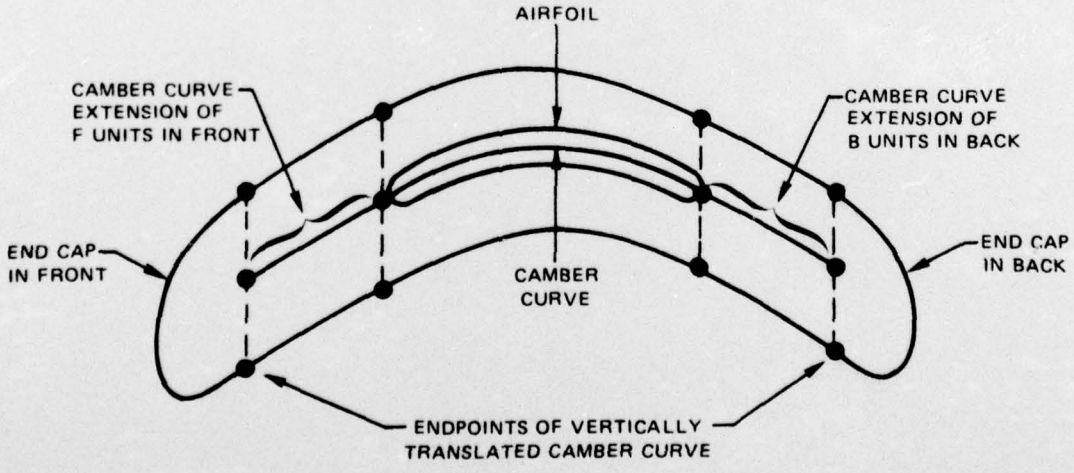

FIGURE 11: THE CONSTITUENT PARTS OF THE OUTER LOOP

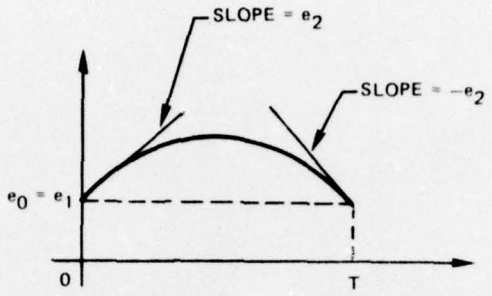

FIGURE 12: QUADRADIC X–COORDINATE

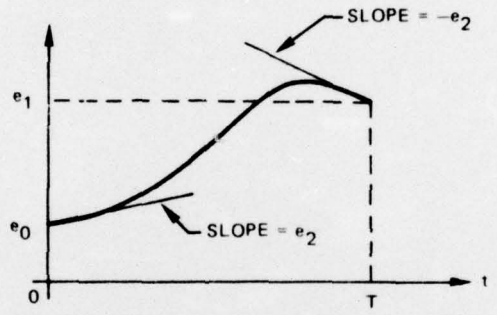

FIGURE 13: CUBIC Y–COORDINATE

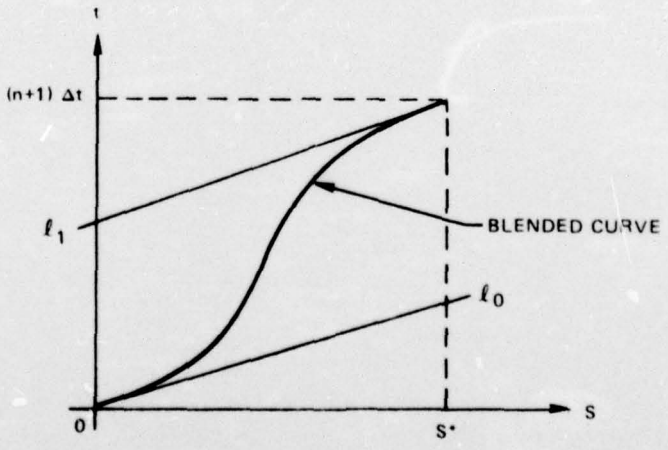

FIGURE 14: A BLEND OF LINES

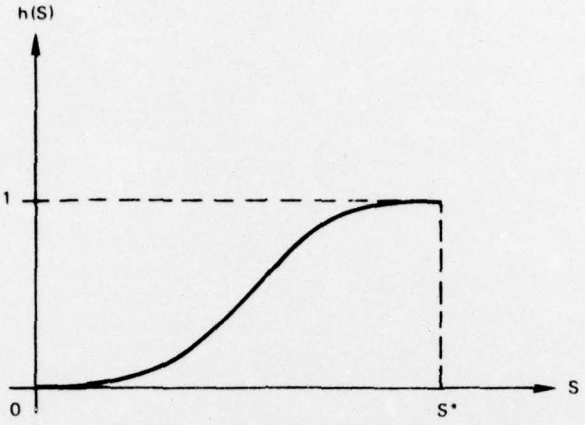

FIGURE 15: HOMOTOPY PARAMETER

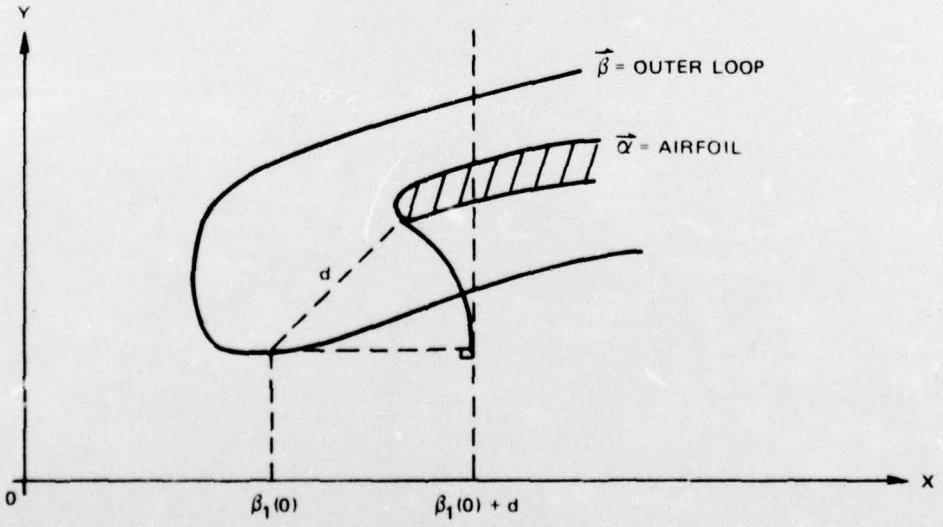

FIGURE 16: REGION OF SEARCH FOR FIRST POINT

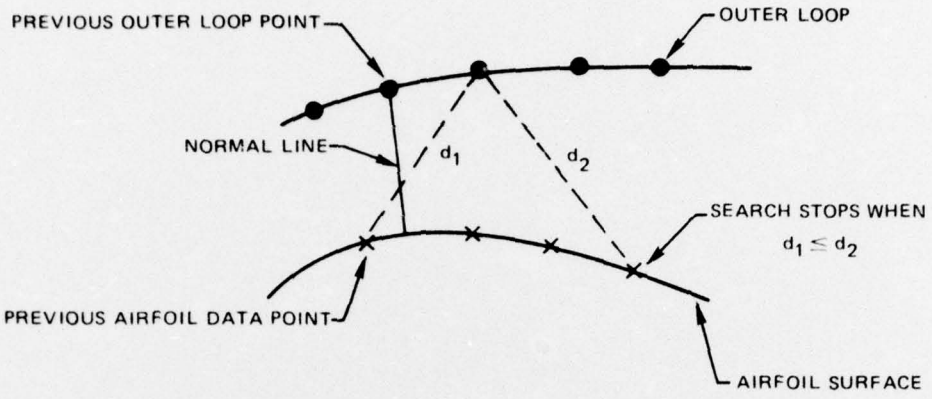

FIGURE 17: CRITERION TO LIMIT THE SEARCH OF EXISTING AIRFOIL DATA

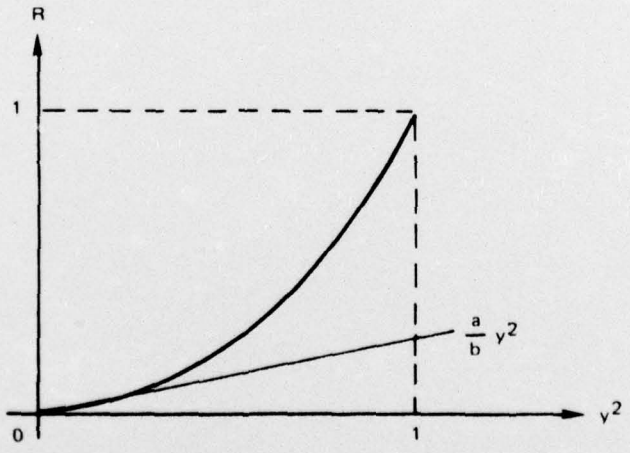

FIGURE 18: DISTRIBUTION FUNCTION FOR THE MESH ALONG THE NORMAL LINES

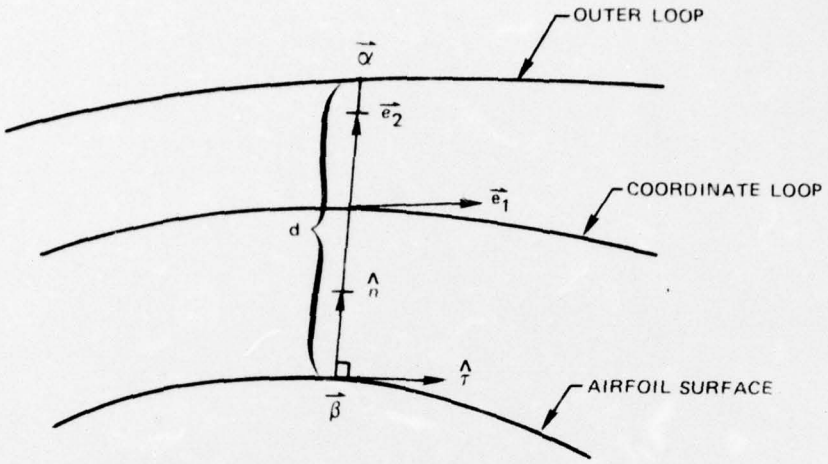

FIGURE 19. INTRINSIC PARAMETERS FOR THE CASCADE COORDINATE SYSTEM

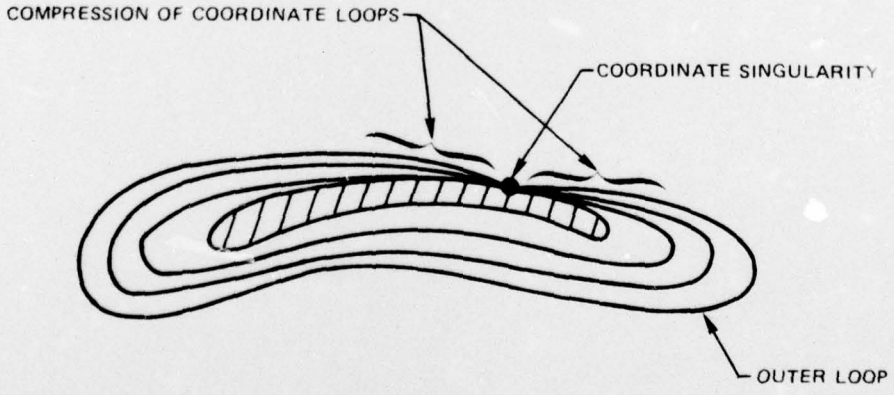

FIGURE 20: COORDINATE SINGULARITY FROM LOCAL COOINCIDENCE OF THE AIRFOIL AND THE OUTER LOOP

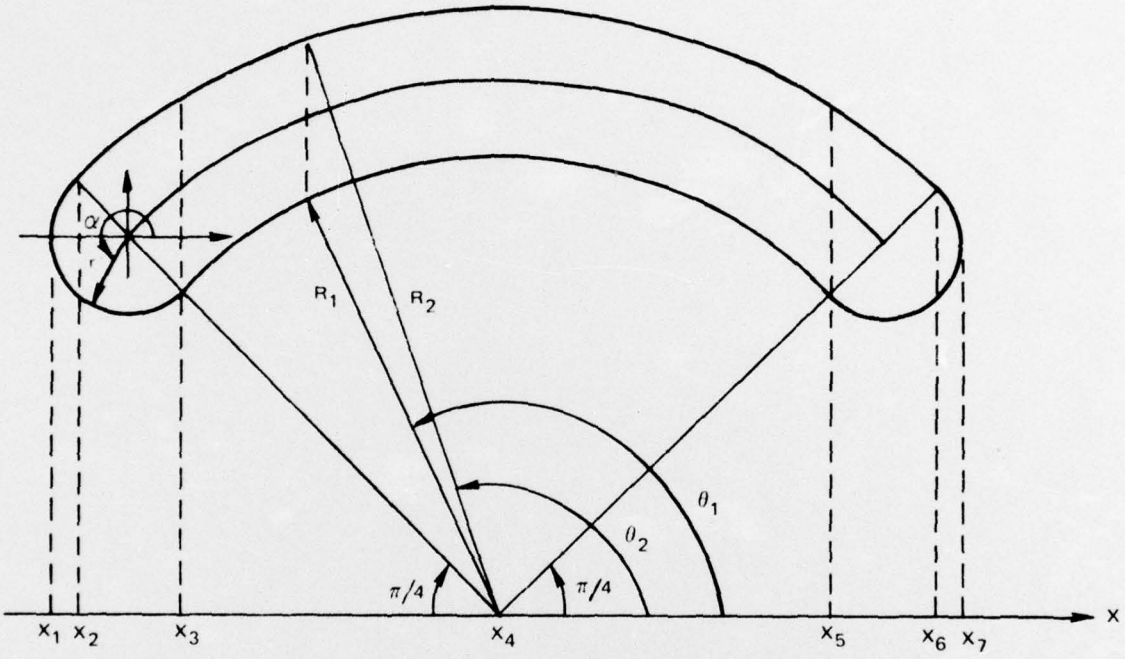

FIGURE 21: CASCADE AIRFOIL FOR TEST CASE

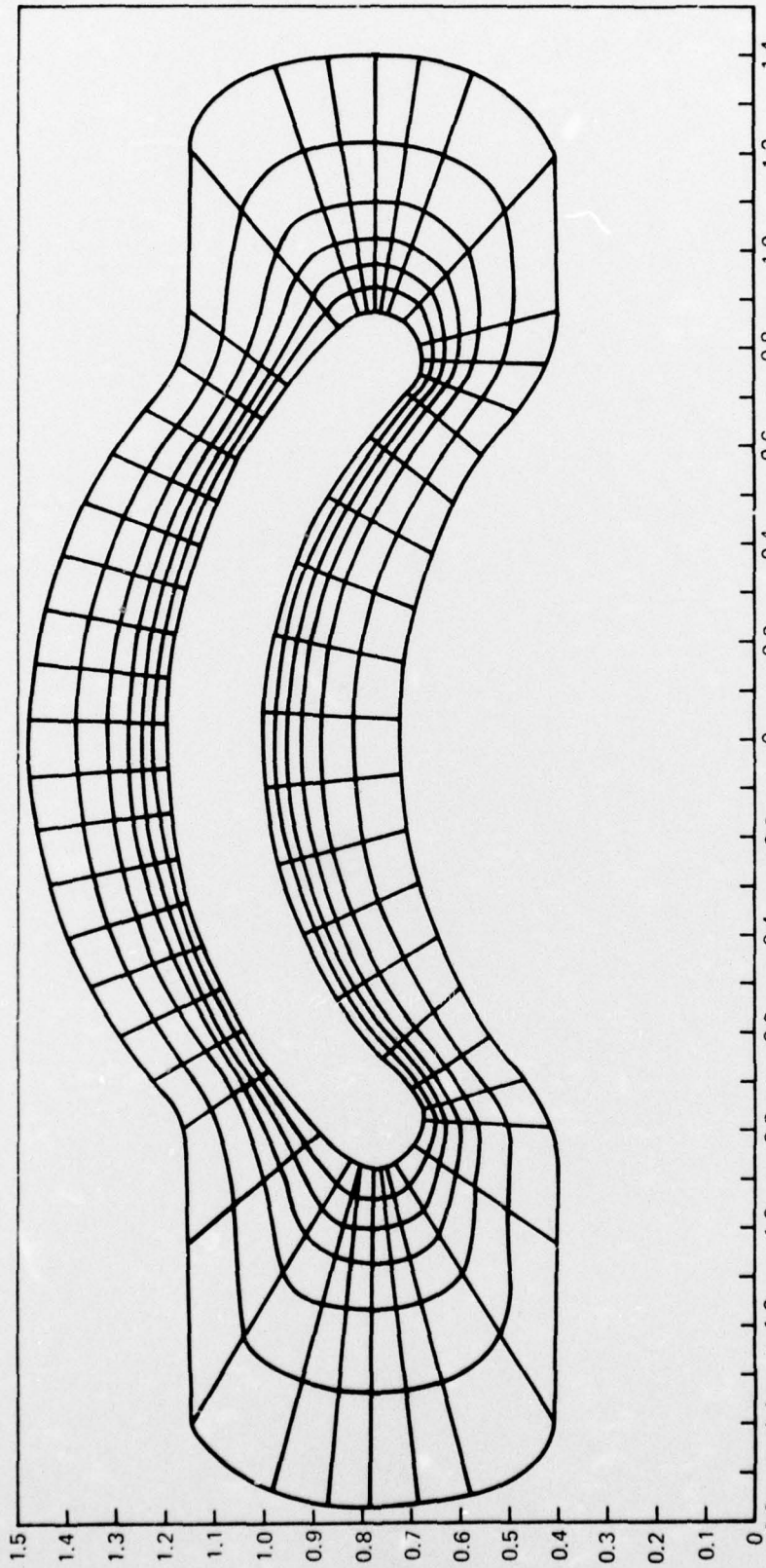

FIGURE 22: COORDINATE SYSTEM GENERATED BY THE CASCADE ALGORITHM

76-04-223-1